

Elektronická žákovská knížka

Electronic pupil book

Zadání bakalářské práce

Student:

Jaroslav Kaštura

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Elektronická žákovská knížka
Electronic Pupil Book

Zásady pro vypracování:

Nynější systémy elektronických třídních knih a žákovských knížek jsou sice propracované, nicméně často jim chybí podpora mobilních zařízení. Úkolem studenta bude vytvořit informační systém skládající se z webového portálu a aplikací pro mobilní zařízení (tablety, mobilní telefony, a pod.). Systém bude umožňovat práci v online i v offline režimu, přičemž data budou synchronizovaná automaticky po připojení k síti.

Úkoly:

1. Prostudujte existující IS elektronických žákovských knížek a zhodnoťte jejich vlastnosti a nedostatky.
2. Navrhněte a naimplementujte informační systém, který bude obsahovat:
 - a) Mobilní aplikaci určenou pro učitele obsahující docházkový systém, rozvrhy, známkování žáků, atd.
 - b) Mobilní aplikaci určenou pro žáky a jejich rodiče obsahující prospěchy, oznámení, informace z dění školy, atd.
 - c) Webový portál poskytující všechny funkce mobilních aplikací a funkce pro administraci celého systému.
3. Porovnejte IS s existujícími systémy a zhodnoťte jeho přínosy.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Peter Chovanec**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 20. dubna 2015



.....

Děkuji tímto vedoucímu bakalářské práce Ing. Petrovi Chovancovi za odbornou pomoc, ochotu a konzultace při realizaci tohoto projektu. Dále bych chtěl poděkovat pedagožce základní školy Františka Formana v Ostravě Mgr. Aleně Saparové za spolupráci při návrhu uživatelského rozhraní IS.

Abstrakt

Cílem této bakalářské práce je vývoj a nasazení informačního systému, takzvané elektronické žákovské knížky, určené pro základní školy. V práci je popsán celý softwarový proces, počínaje výběrem technologií pro jednotlivé aplikace, fyzického návrhu databází, doménového modelu, prezentační vrstvy, komunikace mezi serverem a mobilními aplikacemi a implementaci IS. Dále se práce zabývá testováním výkonu systému a optimalizací datové vrstvy.

Klíčová slova: Informační systém, Elektronická žákovská knížka, optimalizace datové vrstvy, návrh doménového modelu, Java Server Faces, JAVA, MySQL, Jmeter.

Abstract

The aim of this bachelor thesis is the deployment of an information system called Electronic pupil's book designed for elementary schools. The thesis contains a description of the whole software process, starting with the survey of the modern technologies and selection of the optimal one for each of the applications. Afterwards the physical database structure, domain model, presentation layer for each application, the communications between the server and mobile applications are described in detail including implementation of IS. Since, the optimal system performance is required, optimization of the data layer is discussed at the end.

Keywords: Information system, Electronic Pupil Book, optimization of the data layer, domain model design, Java Server Faces, JAVA, MySQL, Jmeter.

Seznam použitých zkratk a symbolů

AJAX	– Asynchronní Javascript a Xml [18] - je v informatice obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich kompletního znovunačítání za pomoci asynchronního zpracování webových stránek pomocí knihovny napsané v JavaScriptu.
GPL	– GNU General Public License [24], GNU GPL (česky „všeobecná veřejná licence GNU“) je licence pro svobodný software, původně napsaná Richardem Stallmanem pro projekt GNU. GPL je nejpopulárnějším a dobře známým příkladem silně copyleftové licence, která vyžaduje, aby byla odvozená díla dostupná pod toutéž licencí.
IS	– Informační systém [5] – propojení informačních technologií a lidských aktivit směřujících k zajištění podpory procesů ve společnosti.
JAX-RS	– Java API for RESTful Services [16] - je API programovacího jazyku Java, které umožňuje vytváření webových služeb.
JS	– JavaScript [23] – programovací jazyk určený pro frontendové aplikace běžící v internetovém prohlížeči.
JSF	– Java Server Faces [12] – technologie byla vyvinuta společností Sun Microsystems, Inc. Je součástí Java 5 Enterprise Edition. Hlavní myšlenkou je možnost čistějšího vývoje profesionálních Web aplikací. Vývojáři definují uživatelský interface pomocí speciálních XML tagů, kterým jsou předávána data k zobrazení / editaci ze standardních Java beanů. Takto je rozdělena Web aplikace čistě na uživatelské rozhraní a aplikační logiku.

ORM

- Objektově Relační Mapování [15] - je programovací technika v softwarovém inženýrství, která zajišťuje automatickou konverzi dat mezi relační databází a objektově orientovaným programovacím jazykem.

Obsah

1	Úvod	6
2	Funkční analýza	7
2.1	Vize	7
2.2	Kdo bude informační systém používat	7
2.3	Seznam funkcí	9
2.4	Nefunkční požadavky	10
3	Použité technologie	12
3.1	Operační systém a aplikační server	12
3.2	Datová vrstva	12
3.3	Aplikační vrstva	13
3.4	Prezentační vrstva	13
4	Datová analýza	16
4.1	Vstupy do informačního systému	16
4.2	Datový model	16
5	Návrh doménového modelu	18
5.1	Definice návrhových vzorů	18
5.2	Návrhové vzory webové aplikace	18
5.3	Návrhové vzory mobilní aplikace pro učitele na platformě Android	19
5.4	Návrhové vzory použité v hybridní aplikaci pro žáky a rodiče	20
6	Testování	23
6.1	Generování syntetických dat	23
6.2	Testování výkonu aplikace	26
6.3	Výsledky testování	31
7	Závěr	36
8	Reference	37
	Přílohy	38
A	Datový slovník	39
B	Testování servisní vrstvy - Virtuální server Wedos	44
C	Testování servisní vrstvy - Raspberry Pi, Aspire One NetBook	51

Seznam tabulek

1	Datový slovník - tabulka SchoolYear	40
2	Datový slovník - tabulka Studygroup	40
3	Datový slovník - tabulka Users	40
4	Datový slovník - tabulka Informations	41
5	Datový slovník - tabulka Sheduleitem	41
6	Datový slovník - tabulka Results	42
7	Datový slovník - tabulka Attendance	42
8	Datový slovník - tabulka ParrentStudent	43
9	Datový slovník - tabulka Studysubject	43

Seznam obrázků

1	Use Case diagram - Administrátor	8
2	Use Case diagram – Učitel, Žák a Rodič	8
3	ER diagram	17
4	Activity diagram - Generování syntetických dat.	25
5	Ukázka injektování přihlašovacích údajů do HTTP GET požadavku v aplikaci JMeter.	27
6	Ukázka performance reports MySQL Workbench.	29
7	Vizuální zobrazení průběhu vykonání dotazu.	30
8	Vizuální zobrazení průběhu vykonání dotazu, příklad chybějícího indexu.	30
9	Vizuální zobrazení průběhu vykonání dotazu, chybějící index doplněn.	31
10	Agregační graf učitelé - VPS server, synchronizace 2 učitelů a 20 žáků současně	33
11	Graf průměrná odezva učitelé - VPS server, synchronizace 2 učitelů a 20 žáků současně	34
12	Agregační graf žáci - VPS server, synchronizace 2 učitelů a 20 žáků současně	45
13	Graf průměrná odezva žáci - VPS server, synchronizace 2 učitelů a 20 žáků současně	45
14	Agregační graf učitelé - VPS server, synchronizace 2 učitelů a 20 žáků současně	46
15	Graf průměrná odezva učitelé - VPS server, synchronizace 2 učitelů a 20 žáků současně	46
16	Agregační graf žáci - VPS server, synchronizace 5 učitelů a 50 žáků současně	47
17	Graf průměrná odezva žáci - VPS server, synchronizace 5 učitelů a 50 žáků současně	47
18	Agregační graf učitelé - VPS server, synchronizace 5 učitelů a 50 žáků současně	48
19	Graf průměrná odezva učitelé - VPS server, synchronizace 5 učitelů a 50 žáků současně	48
20	Agregační graf žáci - VPS server, synchronizace 10 učitelů a 100 žáků současně	49
21	Graf průměrná odezva žáci - VPS server, synchronizace 10 učitelů a 100 žáků současně	49
22	Agregační graf učitelé - VPS server, synchronizace 10 učitelů a 100 žáků současně	50
23	Graf průměrná odezva učitelé - VPS server, synchronizace 10 učitelů a 100 žáků současně	50
24	Agregační graf žáci - RaspberryPI, synchronizace 5 učitelů a 50 žáků současně	52
25	Graf průměrná odezva žáci - RaspberryPI, synchronizace 5 učitelů a 50 žáků současně	52
26	Agregační graf učitelé - RaspberryPI, synchronizace 5 učitelů a 50 žáků současně	53

27	Graf průměrná odezva učitelé - RaspberryPI, synchronizace 5 učitelů a 50 žáků součastně	53
28	Agregační graf žáci - Aspire One NetBook, synchronizace 5 učitelů a 50 žáků součastně	54
29	Graf průměrná odezva žáci - Aspire One NetBook, synchronizace 5 učitelů a 50 žáků součastně	54
30	Agregační graf učitelé - Aspire One NetBook, synchronizace 5 učitelů a 50 žáků součastně	55
31	Graf průměrná odezva učitelé - Aspire One NetBook, synchronizace 5 učitelů a 50 žáků součastně	55

Seznam výpisů zdrojového kódu

1	Příklad HTML tabulky při použití AngularJS a Twitter Bootstrap	14
2	Ukázka Table Data Gateway ve webové aplikaci	18
3	Ukázka využití Single Table Inheritance ve webové aplikaci	19
4	Ukázka Servisní vrstvy ve webové aplikaci	19
5	Ukázka Table Data Gateway na platformě Android	19
6	Ukázka Proactor na platformě Android	20
7	Ukázka Table Data Gateway v hybridní aplikaci	20
8	Ukázka Proactor v hybridní aplikaci	21
9	Ukázka MVC controlleru v hybridní aplikaci	21
10	Ukázka MVC view v hybridní aplikaci	21
11	Ukázka příkazu EXPLAIN	28
12	Ukázka MySQL kódu pro vypsání dotazů s vysokou cenou	28
13	Ukázka MySQL kódu pro vypsání tabulek nad kterými je často prováděn sekvenční průchod.	31

1 Úvod

S nástupem nových médií, sociálních sítí a progresivním rozvojem informačních a komunikačních technologií, byly moderními informačními technologiemi poznamenány téměř všechny lidské činnosti. Těmto novým technologiím se nevyhnulo ani školství, kde tyto moderní informační systémy (dále jen IS) pomalu nahrazují starší neefektivní informační systémy. Jedním s mnoha těchto produktů je elektronická žákovská knížka a elektronická třídní kniha. Tyto IS se těší velké popularitě ze strany rodičů, neboť zde mají přehled o klasifikaci a docházce svých potomků, ale ze strany učitelů bývají často cílem kritiky z mnoha důvodů. Jedním z těchto důvodů je zdvojená práce učitelů, jelikož učitelé často vedou dvojí evidenci klasifikace a docházky, jak „Papírovou“ tak „Elektronickou“.

Problémem dosavadních systémů mohou být také vysoké finanční náklady na pořízení zázemí pro provoz těchto systémů, neboť některé IS jako je například IS firmy Bakaláři s.r.o. Internetová žákovská knížka [32] nebo slovenský IS společnosti ŠEVT a.s. eŠkola [33] musí být instalovány na zařízení školy s komerčním operačním systémem, což vede k vysokým pořizovacím nákladům. Dalšími náklady jsou paušální poplatky za využívání výše uvedených produktů. V případě systému eŠkola se částka pohybuje od €157 od €378 ročně za všechny dostupné komponenty, v závislosti na počtu evidovaných žáků. V případě systému bakaláři se pořizovací cena pohybuje od 2200 Kč do 120000 Kč v závislosti na vybraných komponentách a počtu evidovaných žáků. Navíc cena systému není konečná, jelikož nezahrnuje pravidelné zpoplatněné aktualizace systému.

Alternativou mohou být IS jež nepotřebují zázemí ve školách, neboť jsou provozovány v cloudu. Takový systém nabízí například Brněnská firma Edookit [31]. Tato firma nabízí produkty pro mateřské, základní a střední školy. Přičemž cena za aktivaci produktu se pohybuje od 660 Kč do 3800 Kč a měsíční paušál od 330 Kč od 1900 Kč, dle druhu školy a počtu žáků.

Cílem této bakalářské práce je vypracovat na základě zjištěných poznatků nový IS s ohledem na připomínky osob, jež by jej mohly potencionálně aktivně využívat a s ohledem na snížení pořizovacích a provozních nákladů. Systém musí obsahovat webovou aplikaci pro administrátora, mobilní a webové aplikace pro učitele a mobilní a webové aplikace pro žáky a rodiče s ohledem na zpracování jednotlivých školních agend tj. evidence klasifikace žáků, výchovných opatření, docházky či rozvrhu hodin.

2 Funkční analýza

2.1 Vize

Je zapotřebí zřídit IS, takzvanou Elektronickou žákovskou knížku a Elektronickou třídní knihu pro základní a střední školy. Systém bude sloužit učitelům, žákům a rodičům žáků, kde učitelé budou zapisovat klasifikaci, docházku, informace týkající se školních aktivit a žáci s rodiči budou mít možnost tyto data zobrazit. Dále bude systém poskytovat funkce k organizaci rozvrhů hodin, evidenci žáků, evidenci učitelů a organizaci informací pro žáky a rodiče.

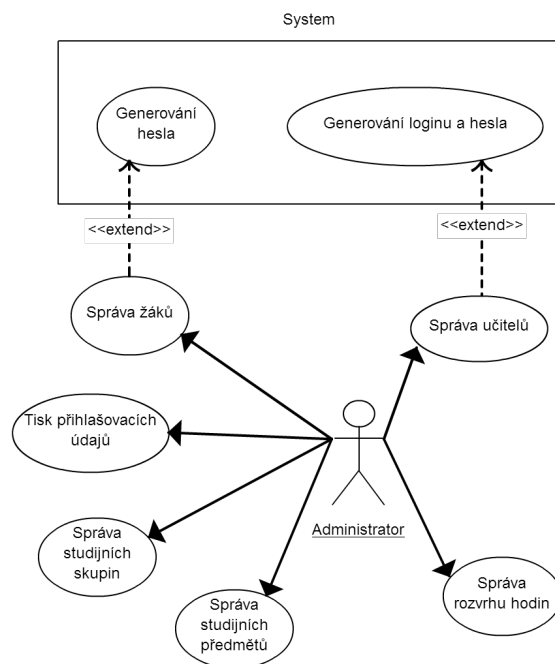
IS bude složen ze tří podsystémů. Prvním podsystémem bude webová aplikace určena koordinátorovi školy neboli administrátorovi systému (dále jen administrátor), kde bude administrátor IS pracovat se seznamy žáků, učitelů, tříd nebo také studijních skupin a bude pomáhat s organizováním rozvrhu hodin. Druhým podsystémem bude webová a mobilní aplikace pro učitele, kde budou učitelé spravovat klasifikaci, výchovná opatření, informace z dění školy a absenci žáků. Třetím podsystémem bude webová a mobilní aplikace pro žáky a rodiče, kde budou mít tyto uživatelé možnost zobrazit klasifikaci, výchovná opatření, absenci a informace z dění školy, přičemž rodiče budou mít navíc možnost absenci omluvit. Učitelé, studenti a rodiče budou mít možnost nahlížet do rozvrhu hodin.

Všechny webové aplikace budou mít responzivní design, mobilní aplikace budou fungovat také offline přičemž po připojení zařízení k internetu bude provedena automatická synchronizace se serverem.

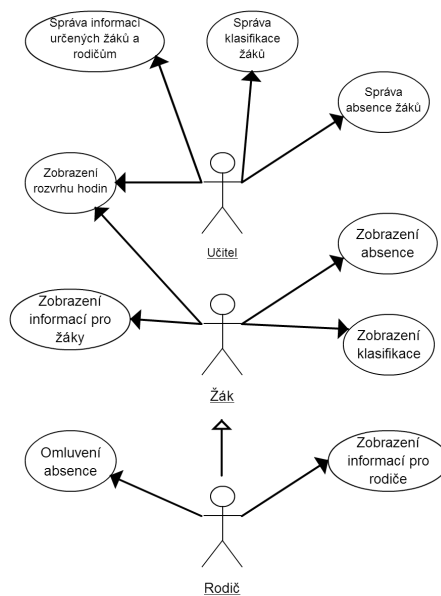
2.2 Kdo bude informační systém používat

Jak bylo uvedeno v předchozí podsekci, IS budou využívat uživatelé v rolích administrátora (viz obrázek č. 1), učitele, žáka a rodiče (viz obrázek č. 2).

- Administrátor systému bude mít k dispozici nástroje ke správě školy. Především tedy evidenci žáků a učitelů, dále evidenci školního roku, evidenci studijních skupin. Dále bude administrátor zodpovídat za editaci studijních předmětů, na základě kterých bude vytvářet rozvrh hodin.
- Učitelé budou mít k dispozici nástroje na prohlížení vlastních rozvrhů hodin, nástroje k evidenci absence žáků, informací pro žáky a rodiče. V neposlední řadě budou mít učitelé k dispozici nástroje na správu klasifikace, to znamená nástroje pro zápis a editaci známek.
- Žáci budou mít k dispozici nástroje, kterými budou moci nahlížet do klasifikace, docházky, informací a rozvrhu hodin.
- Rodiče budou mít k dispozici nástroje, kterými budou moci nahlížet do klasifikace, docházky, informací, rozvrhu hodin a budou moci omlouvat absenci žáků, jimž jsou rodiči.



Obrázek 1: Use Case diagram - Administrátor



Obrázek 2: Use Case diagram – Učitel, Žák a Rodič

2.3 Seznam funkcí

- Správa žáků
 - Přidání žáka
 - Aktualizace žáka
 - Smazání žáka
 - Přiřazení žáka k studijní skupině
 - Seznam žáků
 - Vyhledávání žáků
- Správa učitelů
 - Přidání učitele
 - Aktualizace učitele
 - Smazání učitele
 - Přiřazení učitele k studijní skupině
 - Seznam učitelů
 - Vyhledávání učitelů
- Správa studijních předmětů
 - Přidání studijního předmětu
 - Aktualizace studijního předmětu
 - Smazání studijního předmětu
- Správa studijních skupin
 - Přidání studijní skupiny
 - Aktualizace studijní skupiny
 - Smazání studijní skupiny
- Správa rozvrhu hodin
 - Zobrazení rozvrhu hodin od vybrané studijní skupiny
 - Přidání předmětu a učitele do vybrané hodiny
 - Aktualizace předmětu a učitele ve vybrané hodiny
 - Smazání předmětu a učitele z vybrané hodiny
- Tisk přihlašovacích údajů
 - Výběr uživatelů, kterým budou vygenerovány nové přihlašovací údaje

-
- Generování nových přihlašovacích údajů
 - Tisk nových přihlašovacích údajů
 - Správa klasifikace žáků
 - Zobrazení klasifikace vybraného žáka
 - Přidání hodnocení z vybraného předmětu vybranému žákovi
 - Aktualizace hodnocení z vybraného předmětu vybranému žákovi
 - Smazání hodnocení z vybraného předmětu vybranému žákovi
 - Správa absence žáků
 - Zobrazení absence vybraného žáka
 - Přidání absence vybranému žákovi
 - Aktualizace absence vybraného žáka
 - Omluvení absence vybraného žáka
 - Správa informací pro studijní skupiny, rodiče nebo žáky
 - Zobrazení informací vybraného žáka, rodiče nebo studijní skupiny
 - Přidání informace pro vybraného žáka, rodiče nebo studijní skupinu
 - Aktualizace informace
 - Smazání informace
 - Generování loginu a hesla
 - Generování loginu uživatele - login uživatele bude vytvořen z jeho příjmení tak, že budou systémem zkopírovány první tři znaky příjmení do nového řetězce, ke kterému bude připojeno třímístné pořadové číslo, jehož hodnota bude vypočtená z loginů, které jsou již v systému uloženy a to tak, že bude v databázi vyhledán login se stejnými třemi znaky na začátku řetězce a nejvyšším pořadovým číslem. K tomuto pořadovému číslu bude přičtena hodnota jedna. V případě absence podobného loginu, bude novému uživateli přiděleno pořadové číslo 000.
 - Generování hesla – heslo uživatele bude náhodně vygenerovaný řetězec o velikosti 8 znaků.

2.4 Nefunkční požadavky

Výsledný IS by měl být co nejjednodušší, intuitivní s responsivním designem, neboť se předpokládá využívání této aplikace na různých zařízeních s odlišnými vlastnostmi zobrazovacího zařízení. Předpokládá se využití nejmodernějších opensource technologií, z důvodu efektivity vývoje a snižování nákladů. Zejména pak moderního frameworku

JSF, který splňuje všechny výše uvedené požadavky, databáze MySQL, která je výkonnostně dostačující, má skvělou komunitní podporu a výborné vývojové nástroje a testovací nástroje. Na prezentační vrstvě bude v případě webové aplikace použit frontend framework Primefaces, který v mnoha ohledech předčí komerční produkty. Aplikace bude nasazena na aplikačním serveru Glassfish společnosti Oracle, který bude instalován v operačním systému Linux Ubuntu server, který rovněž splňuje všechny výše zmiňované požadavky.

3 Použité technologie

3.1 Operační systém a aplikační server

3.1.1 Ubuntu

Ubuntu [29] je komunitně vyvíjený operační systém vhodný pro laptopy, stolní počítače i servery. Tento systém je postaven na linuxovém jádře a je velmi důkladně navržen s ohledem na bezpečnost.

3.1.2 Glassfish

GlassFish [20] je aplikační server vyvinutý společností Sun Microsystems, kterou později odkoupila firma Oracle Corporation pro platformu Java EE. Řadí se mezi open source podléhající licencím GPL a CDDL. Jedná se o referenční implementaci, to znamená, že není primárně určena pro provoz aplikací, ale slouží především jako ukázková implementace nových rysů v poslední specifikaci platformy JAVA EE. Vedle open source existuje komerční verze serveru, která se od open source verze téměř neliší, největší rozdíl je především v automatickém stahování aplikací a v podpoře.

3.2 Datová vrstva

3.2.1 MySQL server

MySQL [22] je databázový systém, vytvořený švédskou firmou MySQL AB a nyní vlastněný společností Sun Microsystems, dceřinou společností Oracle Corporation. Jeho hlavními autory jsou Michael „Monty“ Widenius a David Axmark.

MySQL je multiplatformní databáze. Komunikace s ní probíhá – jak už název napovídá – pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními. Pro svou snadnou implementovatelnost (lze jej instalovat na Linux, MS Windows, ale i další operační systémy), výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na databázích používaných v současné době. Velmi oblíbená a často nasazovaná je kombinace Linux, MySQL, PHP a Apache jako základní software webového serveru („technologie LAMP“).

3.2.2 JDBC – Java Database Connectivity

The Java Database Connectivity API [21] je tovární standard pro nezávislou konektivitu mezi velkým množstvím SQL databázových systémů a programovacím jazykem Java. Lze jej využít také pro jiné tabulkové zdroje, jako jsou například tabulkové procesory.

3.2.3 Eclipse Link

EclipseLink [14] je open source projekt firmy Eclipse Foundation. Tento software poskytuje rozšiřitelný framework, který pomáhá Java vývojářům pracovat s mnoha datovými

zdroji, jako jsou například databáze, webové služby, objektové mapování XML a podnikové informační systémy. EclipseLink podporuje mnoho standardů, jako jsou například Java Persistence API, Java Architecture for XML Binding, Java Connector Architecture nebo Service Data Objects.

3.2.4 JPA - Java Persistence API

Java Persistence API [19] je framework programovacího jazyka Java, jenž umožňuje objektově relační mapování, které usnadňuje vývojářům práci s ukládáním objektů do databáze nebo při načítání objektů z databáze. Tento framework, lze používat jak v Javě EE, tak i v Javě SE.

3.3 Aplikační vrstva

3.3.1 JSF - Java server faces

JSF [12] je framework, který umožňuje vývojářům webových aplikací vytvářet uživatelská rozhraní pro JavaServer aplikace. Je podporován aplikačními servery využívajícími Java Enterprise Edition (Java EE).

JSF zjednodušuje vytváření webových aplikací poskytnutím standardní sady nástrojů pro vytváření uživatelského rozhraní. Například místo kódování webového formuláře v HTML může vývojář zavolat jednoduchou JSF funkci, která vytvoří požadovaný formulář. Další funkce potom může zpracovat data vložené uživatelem. Tyto funkce jsou zpracovány na serveru a zpracovaná data jsou odeslána zpět do klientova prohlížeče. Mezi nejvýznamnější benefity této technologie patří zejména jednoduché vkládání objektů do webových stránek. Vzhledem k tomu, že je Java kód zpracován na serveru, je podoba generovaných webových objektů konzistentní napříč několika webovými stránkami. Kromě toho jsou JSF komponenty testovány na různých platformách, a proto fungují skvěle na většině webových prohlížečů.

3.3.2 JAX-RS - Java API for RESTful Services

Java API for RESTful Services [16] je API programovacího jazyka Java, které poskytuje podporu při vytváření webových služeb v souladu s architektonickými vzory Representation State Transfer (Rest) [17]. JAX-RS používá anotace k zjednodušení vývoje a nasazením webových služeb, klientů a koncových zařízení.

3.4 Prezentační vrstva

3.4.1 HTML5

HTML5 [25] je v informatice verze značkovacího jazyka HTML sloužícího pro tvorbu webových stránek.

3.4.2 Prime faces

PrimeFaces [13] je sada open source komponent uživatelského rozhraní určeného pro aplikace na bázi JSF, a tato sada je vytvořena tureckou firmou PrimeTek.

3.4.3 Java Script

JavaScript [23] je multiplatformní, objektově orientovaný skriptovací jazyk. Zpravidla se používá jako interpretovaný programovací jazyk pro webové aplikace, mnohdy vložený přímo do HTML kódu stránky. Jsou jím obvykle ovládány různé interaktivní prvky uživatelského rozhraní nebo animace, efekty a podobně. Jeho syntaxe patří do rodiny jazyků C, C++ nebo Java. Tento programovací jazyk nemá kromě syntaxe téměř nic společného s programovacím jazykem Java, slovo Java figuruje v tomto názvu pouze z marketingových důvodů.

3.4.4 Twitter Bootstrap

Twitter Bootstrap [27] je velmi populární HTML, CSS a Javascript framework pro vývoj responsivních nebo mobilních aplikací na webu. S tímto frameworkem je velmi jednoduché vytvořit aplikaci založenou na jednom kódu, která bez komplikací funguje na mobilním telefonu, tabletu nebo stolním počítači pomocí CSS dotazů.

3.4.5 AngularJS

AngularJS [26] je open source udržovaný společností Google a komunitou individuálních vývojářů, vhodný pro tvorbu single-page aplikací. Jeho cílem je zjednodušit vývoj a testování aplikací poskytnutím frameworku na bázi MVC na straně klienta spolu s komponentami běžně používanými v čistě internetových aplikacích.

```
<table class="table table-striped">
  <tr>
    <th>Nepritomen od</th><th>Nepritomen do</th><td>Omluven</td><td ng-show="{{
      loggedUserIsParrent}}">Akce</td>
  </tr>
  <tr ng-repeat="item_in_attendance" class="{{item.excused_===_true'_?_'success'_:_danger
    '}}">
    <td>{{item.start | date:'dd.MM.yyyy_HH:mm'}}</td>
    <td>{{item.end | date:'dd.MM.yyyy_HH:mm'}}</td>
    <td>{{item.excused === 'true' ? 'ANO' : 'NE'}}</td>
    <td ng-show="{{loggedUserIsParrent}}"><button ng-show="{{loggedUserIsParrent}}" ng-
      disabled="{{item.excused_===_true'}}" class="btn btn-default" ng-click="excuseClick(
        item.id)">Omluvit</button></td>
  </tr>
</table>
```

Výpis 1: Příklad HTML tabulky při použití AngularJS a Twitter Bootstrap

Knihovna funguje tak, že nejdříve přečte celou HTML stránku obsahující speciální atributy, jenž jsou interpretovány jako direktivy, které říkají angularu, aby propojil vstupní

nebo výstupní části stránky s modelem reprezentovaným klasickými JavaScriptovými proměnnými. Hodnoty těchto JavaScriptových proměnných mohou být manuálně nastaveny kódem nebo je lze získat ze statických nebo dynamických JSON zdrojů.

3.4.6 Android

Android [28] je open source mobilní operační systém, postavený na linuxovém jádře. Jádro tohoto operačního systému bylo navrženo pro fungování na různém hardware, především na mobilních zařízeních, kde vývojáři museli brát ohled na omezené množství paměti a nízký výpočetní výkon. Systém lze také použít bez ohledu na použitý chipset, velikost či rozlišení obrazovky. Platforma Android má v době psaní této bakalářské práce 80% podíl na trhu.

4 Datová analýza

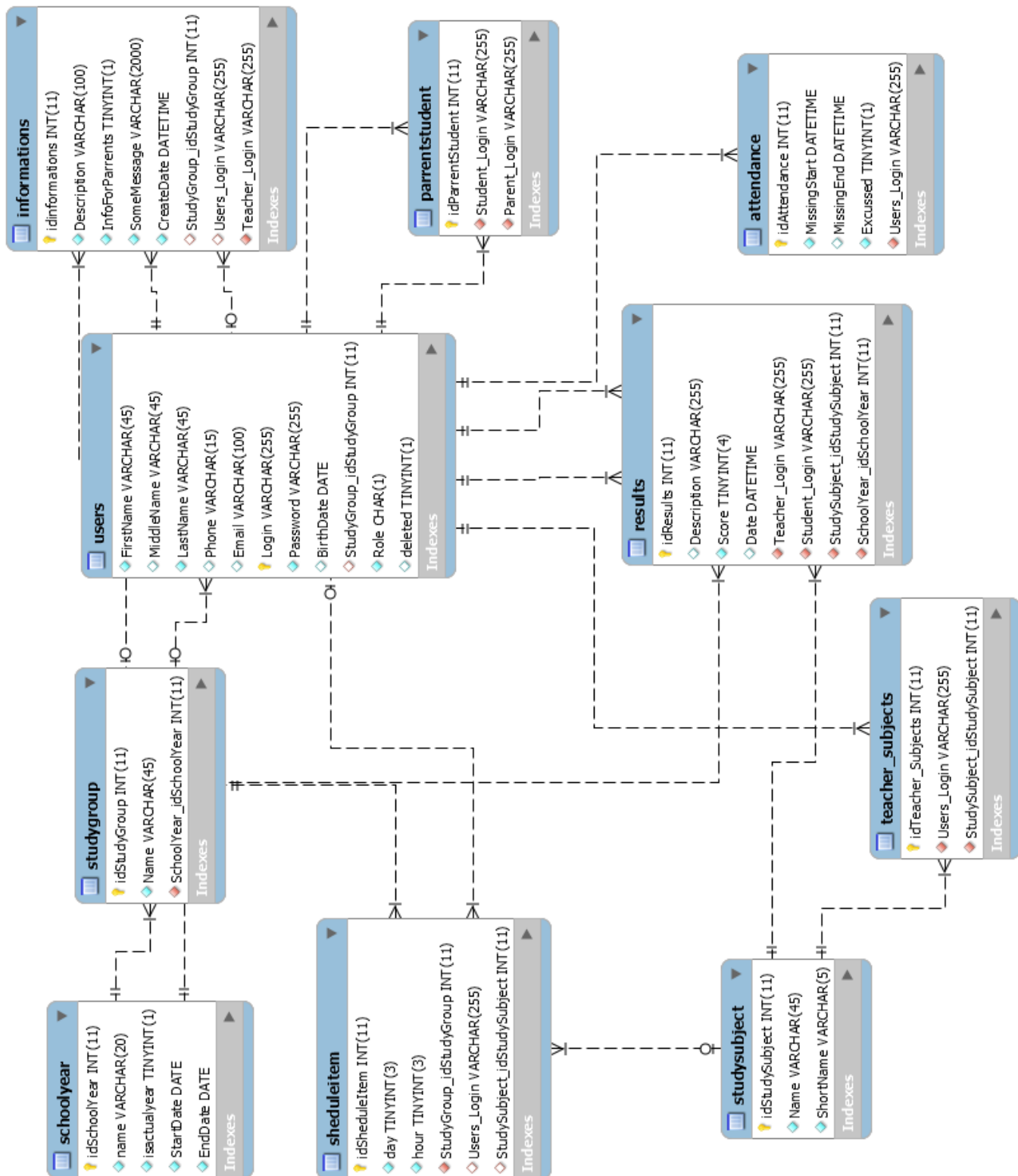
4.1 Vstupy do informačního systému

Vstupy do informačního systému rozumíme informace, které budou systémem zpracovávány a které budeme shromažďovat v databázi. S těmito údaji budu dále v systému pracovat a rozvíjet je v dalších funkcích.

- Žák
Identifikace žáka, jméno a příjmení, případně prostřední jméno, adresa, telefon, email, rodné číslo, datum narození a přístupové heslo.
- Rodič
Identifikace rodiče a přístupové heslo.
- Učitel
Identifikace učitele, přístupové heslo, jméno a příjmení, případně prostřední jméno, adresa, telefon, email, rodné číslo a datum narození.
- Studijní předmět
Název předmětu a zkratka, která se bude zobrazovat v rozvrhu hodin.
- Školní rok
Jméno školního roku, informace o aktuálnosti školního roku, datum začátku a konce.
- Studijní skupina
Název studijní skupiny. Tato studijní skupina bude vázána k výše zmíněnému školnímu roku, dále se do studijních skupin budou zařazovat žáci.
- Rozvrh hodin
Rozvrh hodin bude vázán ke studijní skupině. Do tohoto rozvrhu bude možné přidávat studijní předměty a učitelé, kteří je budou vyučovat.
- Docházka
Začátek absence, případný konec a případná omluvenka absence.
- Klasifikace
Identifikace učitele, který bude klasifikovat žáka, jehož identifikace zde bude rovněž uložena. Krom identifikace bude tato položka obsahovat číslo v rozsahu 0 – 100, které bude představovat procentuální hodnocení žáka a referenci na předmět, ze kterého byl žák hodnocen.

4.2 Datový model

Na obrázku č.3 je zobrazen entitně relační model IS. Důkladná analýza se nachází v příloze A – Datový slovník.



Obrázek 3: ER diagram

5 Návrh doménového modelu

Při vývoji tohoto IS jsem použil řadu návrhových vzorů z knihy Martina Fowlera *Patterns of Enterprise Application Architecture* [5] a v této kapitole rozeberu využití těchto vzorů v jednotlivých aplikacích.

5.1 Definice návrhových vzorů

- Data Transfer Object - je objekt, který přenáší data mezi procesy.
- Table Data Gateway - slouží jako brána k databázovým tabulkám.
- Single Table Inheritance - Představuje hierarchii dědičnosti tříd, které jsou mapovány do jedné tabulky databáze, jenž obsahuje sloupce pro všechny třídy v hierarchii a jeden sloupec, kterým rozlišuje, jaká třída bude instanciována.
- Servisní vrstva - definuje hranice aplikace jako vrstvu, která poskytuje sadu dostupných operací a koordinuje reakce aplikace v každé operaci.
- Proactor - slouží pro řízení událostí, při běhu dlouho trvajících asynchronních aktivitě.

5.2 Návrhové vzory webové aplikace

Jak již bylo uvedeno, webová aplikace je postavená na technologii JSF [12], která již v sobě obsahuje velké množství návrhových vzorů, které jsou před vývojářem skryté a nebudou je v této práci popisovat.

- Data Transfer Object - tento návrhový vzor jsem použil pro předávání dat mezi vrstvami aplikace, jak je uvedeno v dalších příkladech.
- Table Data Gateway - tento návrhový vzor jsem použil pro komunikaci JSF aplikace s databázovým systémem. Níže je ukázka metody, která přijímá dva argumenty na základě kterých odešle databázovému systému SQL dotaz a ten po zpracování dotazu odešle data zpět do aplikace, která dále vytvoří kolekci Data Transfer objektů a tu vrátí metodě, kterou byla volána.

```
public List<Results> getStudentResults(String login, int StudySubjectID){
    return em.createNativeQuery("select * from results WHERE Student_Login = ?login
        AND SchoolYear_idSchoolYear = ?syld AND StudySubject_idStudySubject = ?
        StudySubjectID", Results.class)
        .setParameter("login", login)
        .setParameter("syld", getActualSchoolYear())
        .setParameter("StudySubjectID", StudySubjectID)
        .getResultList();
}
```

Výpis 2: Ukázka Table Data Gateway ve webové aplikaci

- Single Table Inheritance - single table inheritance jsem použil v ORM k mapování typu uživatele v databázi. V datové vrstvě je jedna třída Users, jejíž vlastností dědí třídy Admin, Teacher, Parrent a Student, přičemž až v potomku třídy Users je definován atribut Role, což mi zjednodušilo práci při načítání uživatelů z databáze, neboť jsem mohl při zjišťování role uživatele použít funkci instanceof, jak je uvedeno v příkladu níže, kde můžete vidět metodu, kterou systém využívá k přihlašování.

```

public void login() throws IOException{
    UserObject.setLogin(user);
    UserObject.setPassword>Password);
    Users loggedInUser = sb.doLogin(UserObject);
    if (loggedInUser instanceof Admin) {
        UserObject = loggedInUser;
        AdminUser = true;
    } else if (loggedInUser instanceof Teacher) {
        UserObject = loggedInUser;
        TeacherUser = true;
    } else{
        FacesContext.getCurrentInstance().addMessage(null, new FacesMessage(
            FacesMessage.SEVERITY_ERROR, "Chyba", "Prihlaseni_se_nezdarilo,
            zkontrolujte_prosim_prihlasovaci_udaje"));
    }
}

```

Výpis 3: Ukázka využití Single Table Inheritance ve webové aplikaci

- Servisní vrstva - tuto servisní vrstvu jsem vytvořil pomoci Java API for RESTful Services JAX-RS. Jako data transfer objekty jsem pro tento účel vytvořil speciální sadu tříd, z nichž jsem instancoval objekty vhodné pro přenos dat na další platformy.

```

@GET
@Path("/{login}/{password}")
@Consumes({"application/xml", "application/json"})
public List<servicesDTO.Results> getResult(@PathParam("login") String login,
    @PathParam("password") String password){
    return TeachersSB.getResultTeacher(login, password);
}

```

Výpis 4: Ukázka Servisní vrstvy ve webové aplikaci

5.3 Návrhové vzory mobilní aplikace pro učitele na platformě Android

- Data Transfer Object - tento návrhový vzor jsem použil pro předávání dat mezi vrstvami aplikace, jak je uvedeno v dalších příkladech.
- Table Data Gateway - tento návrhový vzor jsem použil pro přístup aplikace k datům uloženým v lokální databázi přístroje.

```

public Result getResult(int id){
    SQLiteDatabase db = getReadableDatabase();
}

```

```

Cursor cursor = db.query(Utils.TABLE_RESULTS, new String[] {Utils.RESULTS_KEY_ID,
    Utils.RESULTS_DESCRIPTION, Utils.RESULTS_SCORE, Utils.RESULTS_DATE,
    Utils.RESULTS_STUDY_SUBJECT_ID, Utils.RESULTS_STUDENT_LOGIN}, Utils.
    RESULTS_KEY_ID + "=?", new String[] {String.valueOf(id)}, null, null, null, null);
if (cursor != null) {
    cursor.moveToFirst();
}
Result sy = new Result(Integer.parseInt(cursor.getString(0)), cursor.getString(1),
    Integer.parseInt(cursor.getString(2)), cursor.getString(3), Integer.parseInt(cursor.
    getString(4)), cursor.getString(5), cursor.getString(6));
db.close();
return sy;
}

```

Výpis 5: Ukázka Table Data Gateway na platformě Android

- Proactor - tento návrhový vzor jsem použil pro připojení aplikace k servisní vrstvě webové aplikace.

```

public class UkazkovaTrida extends AsyncTask<String, Void, String> {
    public DownloadWebpageTask(MainActivity context) {
        // Klasicky konstruktor
    }
    @Override
    protected String doInBackground(String... urls) {
        // Tato metoda provadi nejakou cinnost na pozadi.
    }
    @Override
    protected void onPostExecute(final String result) {
        // Kod který se provede po dokončení operace na pozadí
    }
}

```

Výpis 6: Ukázka Proactor na platformě Android

5.4 Návrhové vzory použité v hybridní aplikaci pro žáky a rodiče

Definice 5.1 Hybridní aplikace, je aplikace, která dokáže využít hardwarových schopností mobilního zařízení a zároveň je použitelná na několika platformách.

- Data Transfer Object - tento návrhový vzor jsem použil pro předávání dat mezi vrstvami aplikace, jak je uvedeno v dalších příkladech.
- Table Data Gateway - tento návrhový vzor jsem použil pro přístup aplikace k datům uloženým v lokální databázi prohlížeče.

```

function getDBResults($scope){
    var db = getDB();
    var studySubjects = {};
    db.transaction(function (tx) {
        tx.executeSql("SELECT _id, _name _FROM _studySubjects", [],

```

```

        function (tx, results) {
            for (var i = 0; i < results.rows.length; i++){
                studySubjects[i] = results.rows.item(i);
                buildResultsAccordion($scope, results.rows.item(i), i);
            }
        }, null);
    });
};
}

```

Výpis 7: Ukázka Table Data Gateway v hybridní aplikaci

- Proactor - tento návrhový vzor jsem použil pro připojení aplikace k servisní vrstvě webové aplikace.

```

$http.get(URL + "Students/" + login + "/" + password + "/results")
    .success(function(data){
        setDBResults(data);
        createBackColor(data);
        $scope.results = data;
    }).error(function () {
        getDBResults($scope)
    });
}

```

Výpis 8: Ukázka Proactor v hybridní aplikaci

- MVC - tento návrhový vzor jsem použil pro zobrazení dat z modelu v prezentační vrstvě hybridní aplikace.

```

home.directive('results', function () {
    return {
        restrict: 'E',
        templateUrl: 'results.html',
        controller: function ($scope, $http) {
            $scope.results = {};
            $http.get(URL)
                .success(function (data) {
                    $scope.results = data;
                }).error(function () {
                    getDBResults($scope)
                });
        },
        controllerAs: 'resCtrl'
    };
});

```

Výpis 9: Ukázka MVC controlleru v hybridní aplikaci

```

<div class="container" >
    <h1>klasifikace</h1>
    <accordion>

```

```
<accordion-group heading="{{result.name}}" ng-repeat="result_in_results" class="
  text-center">
  <table class="table table-striped">
    <tr ng-repeat="i_in_result.results" class="{{i.classss}}">
      <td>{{i.desc}}</td>
      <td>{{i.score}}</td>
    </tr>
  </table>
</accordion-group>
</accordion>
</div>
```

Výpis 10: Ukázka MVC view v hybridní aplikaci

6 Testování

Cílem testování bylo odhalit nedostatky v datové vrstvě a zjistit požadovaný výkon serveru pro předem stanovený počet uživatelů. Testování jsem rozdělil do těchto 2 kroků: Vygenerování syntetických dat a měření výkonu.

6.1 Generování syntetických dat

Generování dat je možné provádět pomocí open source a komerčních aplikací nebo vytvořením vlastního generátoru. V následujících podkapitolách popíšu vybrané generátory dat, jejich klady a zápory.

6.1.1 EMS Data Generator for MySQL

EMS generátor dat pro MySQL [9] je komerční nástroj pro generování zkušebních dat do databáze MySQL, který generuje realistické údaje pro účely testování databáze.

- Výhody
 - Automatické mapování databáze
 - Lze použít pro libovolnou aplikaci, která využívá MySQL databázi
 - Cena aplikace pro nekomerční použití cca \$50 a pro komerční použití cca \$120
- Nevýhody
 - Nelze používat regulární výrazy
 - V trial verzi lze vygenerovat najednou pouze 100 řádků dat na databázi
 - Nelze generovat přihlašovací údaje uživatelů do souboru

6.1.2 Datanamic Data Generator for MySQL

Datanamic generátor dat pro MySQL [10] je komerční nástroj pro generování zkušebních dat do databáze MySQL, který generuje realistické údaje pro účely testování databáze.

- Výhody
 - Automatické mapování databáze
 - Možnost použití regulárních výrazů
 - Lze použít pro libovolnou aplikaci, která využívá MySQL databázi
 - V trial verzi lze vygenerovat najednou 25000 řádků dat
- Nevýhody
 - Cena aplikace cca \$300
 - Nelze generovat přihlašovací údaje uživatelů do souboru

6.1.3 Opensource datové generátory

Kromě velkých komerčních datových generátorů existuje také velké množství opensource datových generátorů. V případě MySQL jde především o online nástroj Generate Data [11]. Byť tento nástroj generuje data, neumí automaticky mapovat databázi a z toho vyplývá, že byl tento generátor pro můj projekt téměř nepoužitelný, neboť bych musel vyvinout velké úsilí, abych zde vytvořil celý model datové vrstvy.

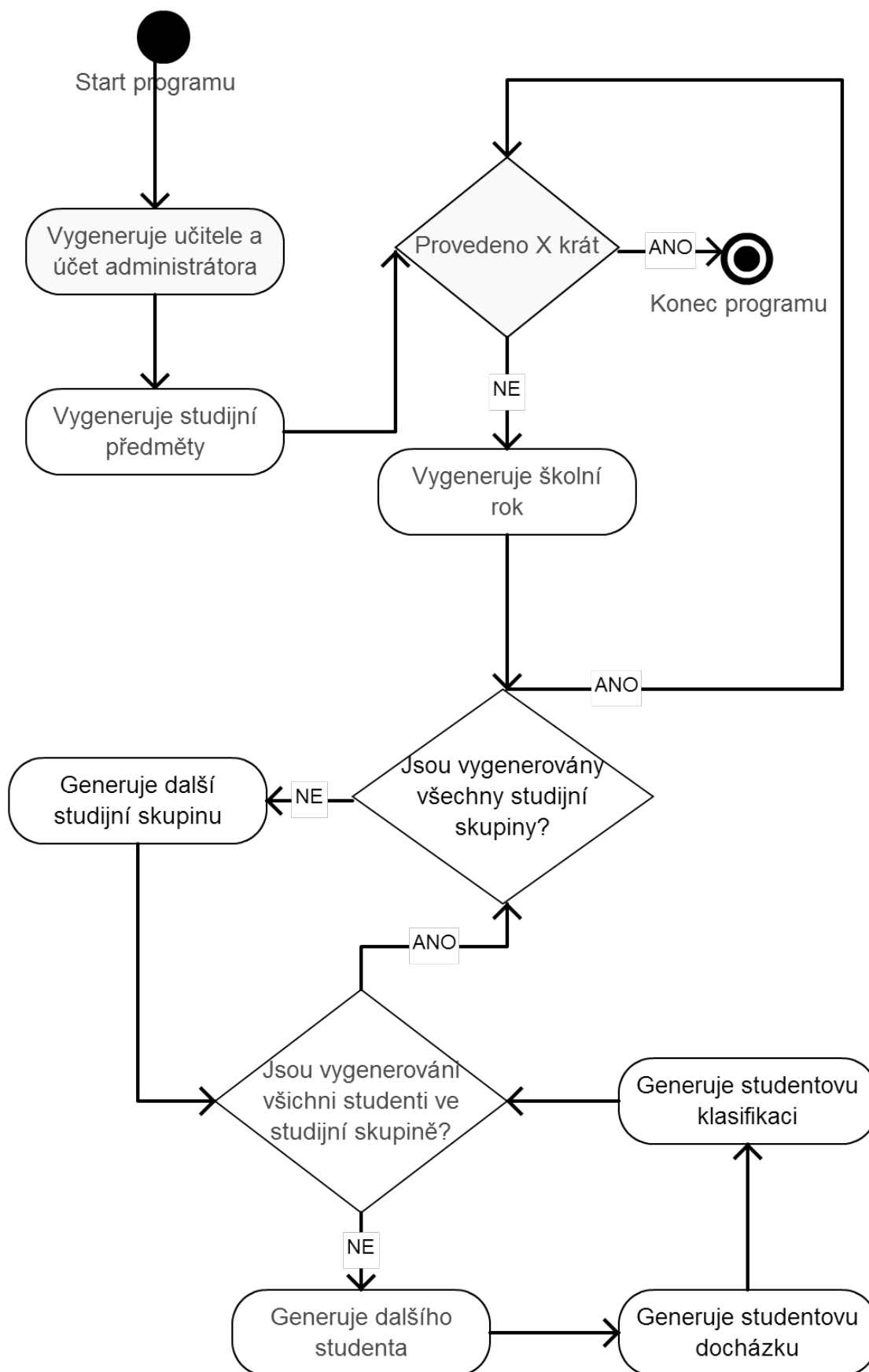
- Výhody
 - Aplikace je zdarma
 - Lze použít pro libovolnou aplikaci, která využívá MySQL databázi
- Nevýhody
 - Nelze použít automatické mapování databáze
 - Nelze používat regulární výraz
 - Nelze generovat přihlašovací údaje uživatelů do souboru

6.1.4 Vlastní generátor dat

Největším problémem, při použití výše zmíněných generátorů, byla nekonzistence generovaných dat. Například ve studijní skupině mohl generátor vytvořit například 500 žáků, zatímco v jiné studijní skupině nebyl žák žádný, což by mohlo způsobit neobjektivní výstupy testování. Proto jsem se rozhodl vytvořit vlastní generátor, kterým jsem databázi naplnil libovolným množstvím konzistentních dat.

Generátor funguje následovně. V prvním kroku vygeneruje předdefinovaný počet školních roků. V následném kroku vygeneruje pro každý školní rok předem definovaný počet studijních skupin a přiřadí je k tomuto školnímu roku tak, jak je uvedeno v Aktivitě diagramu na obrázku č.4. Tímto způsobem byla databáze naplněna libovolným množstvím konzistentních dat, nad kterým bylo možné spouštět testy i samotnou aplikaci. Zároveň při generování uživatelů byly přihlašovací údaje zapisovány do souboru v csv formátu, což jsem dále využil při zátěžových testech v aplikaci Jmeter.

- Výhody
 - Aplikace je zdarma
 - Generování konzistentních dat, nad kterými lze bez problému spustit aplikaci
 - Generování přihlašovacích údajů uživatelů do souboru, což lze dále využít při testování
- Nevýhody
 - Nelze použít automatické mapování databáze
 - Časová náročnost
 - Nelze použít pro libovolnou aplikaci



Obrázek 4: Activity diagram - Generování syntetických dat.

6.2 Testování výkonu aplikace

Testování výkonu aplikace se provádí z důvodu získání reálných informací týkajících se připravenosti aplikace k nasazení a sledování aplikace na straně serveru. To se provádí simulací zatížení v reálných podmínkách, aby se posoudilo, zda aplikace bude podporovat očekávané zatížení. Takovým testováním dokážeme včas identifikovat slabé místa v aplikaci a zaručíme požadovanou dostupnost služeb po nasazení aplikace. Aplikaci jsem testoval níže uvedenými testy.

- Zátěžové testy [30]

Zátěžové testy simulují předpokládané chování uživatelů na základě předem definovaných reálných scénářů použití aplikace. Zátěžové testy ověřují, jestli je aplikace schopna obsloužit požadovaný počet návštěvníků a zpracovat potřebný počet dotazů.

- Stress testy [30]

Stress testy mají za účel ověřit robustnost aplikace. Při Stress testech generujeme na aplikaci zatížení, které postupně zvyšujeme a čekáme, až aplikace spadne. Takto jsme schopni zjistit maximální možnosti aplikace a do budoucna budeme vědět, jak se aplikace bude chovat v případech, kdy její zatížení bude vyšší, než je zatížení předpokládané.

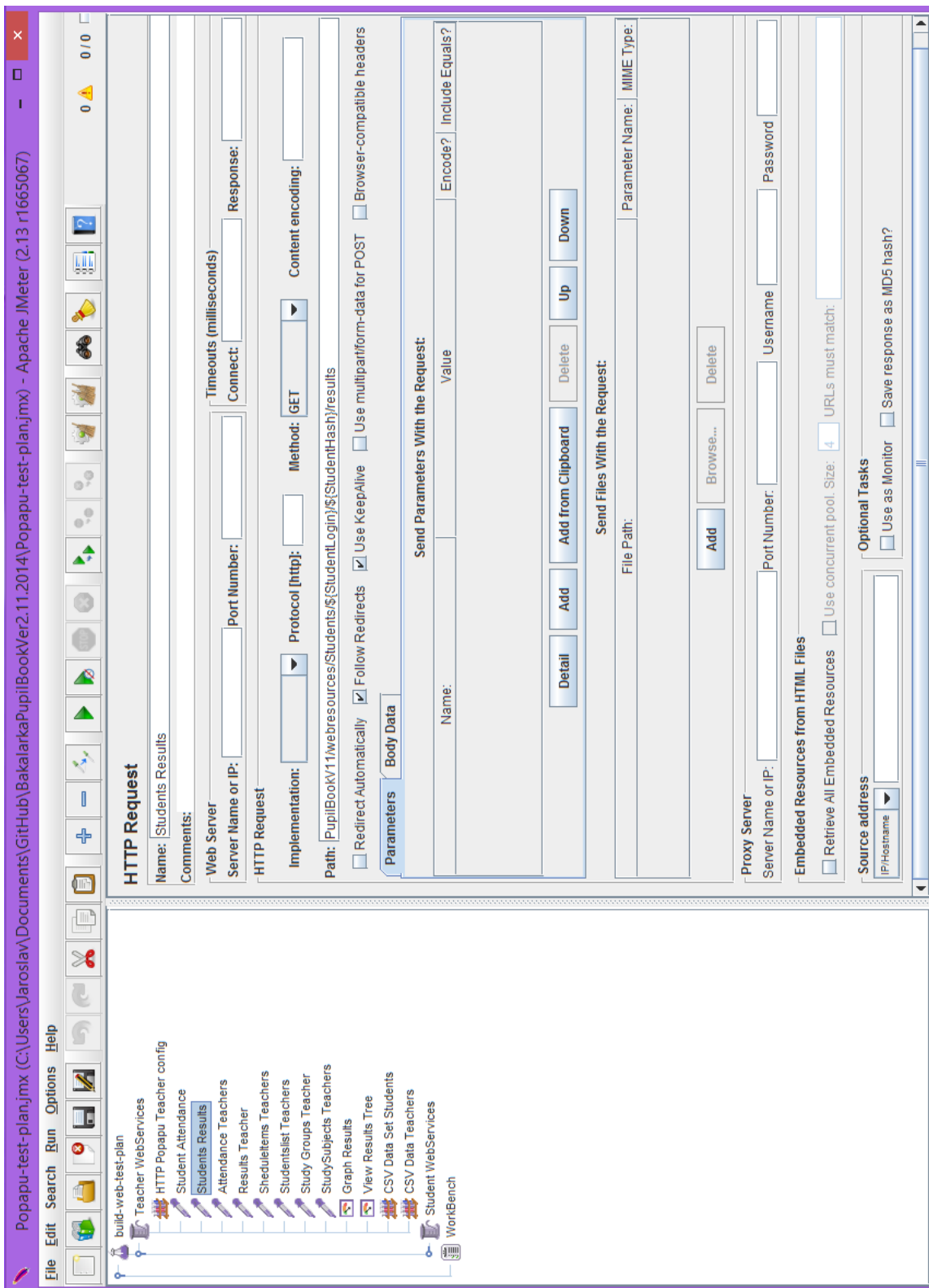
Zátěžové a Stress testy aplikace jsem prováděl v aplikaci Jmeter. V tomto nástroji jsem vytvořil takzvaný scénář, ve kterém mobilní aplikace provádí synchronizaci se serverem, tak jak je uvedeno v doménovém modelu. K tomuto testu jsem využil soubory obsahující přihlašovací údaje všech uživatelů v databázi, které vygeneroval výše uvedený program na generování dat v datové vrstvě. Aplikace Jmeter zpracovala tyto soubory a přihlašovací údaje jednotlivých uživatelů injektovala dle nastavení do HTTP GET požadavku viz. obrázek č. 5.

6.2.1 Analýza datové vrstvy

U analýzy datové vrstvy je cílem zejména nalezení chybějících indexů, vyhledávání nevyužitých indexů a optimalizaci SQL dotazů [4].

Definice 6.1 *Indexy jsou datové struktury, které se využívají při ukládání dat, pro rychlé nalezení řádku v datovém souboru.*

Indexy jsou velmi důležité pro dobrý výkon databázového systému. Navíc se zvětšujícím se objemem dat v databázi jejich důležitost narůstá. Malé, nebo málo zatížené databáze fungují dobře i bez indexů, ovšem pokud databáze začne růst, její výkon může klesnout velmi rychle. Dále je třeba brát ohled na míru selekce každého dotazu. Budeme-li uvažovat dotaz s nízkou selekcí, například 80 000 záznamů z 100 000 možných, potom index zpomalí vyhodnocení takového dotazu. Samozřejmě, špatné indexy nemohou za



Obrázek 5: Ukázka injektování přihlašovacích údajů do HTTP GET požadavku v aplikaci JMeter.

všechny problém spojené s přetížením IS, nicméně optimalizace indexů je nejlepší způsob jakým lze vylepšit výkon dotazování na SQL server a proto jsem tomuto problému věnoval celou podkapitolu.

Analýzu datové vrstvy jsem prováděl v nástroji MySQL Workbench. V tomto nástroji jsou pro testování předpřipravené funkce takzvané Performance Reports (viz. obrázek č.6), kde jsou předdefinovány dotazy, jejichž výsledky lze použít k analýze dotazů. Tyto dotazy lze zapisovat manuálně.

Vzhledem k charakteru aplikace je jasné, že budou v drtivé většině případů používat dotazy s vysokou mírou selekce a proto jsem největší důraz kladl na vyhledání chybějících indexů. Po nalezení dotazu s vysokou cenou vykonání, lze k dotazu vygenerovat Query Execution Plan, kterým lze najít slabé místo nebo v nástroji MySQL Workbench kliknout na ikonu Show Execution plan (viz. obrázek č.7) a nástroj graficky zobrazí plán dotazu včetně ceny jednotlivých operací. V případě absence indexu v některém z poddotazu, nástroj MySQL Workbench upozorní na tuto skutečnost vybarvením atributu červenou barvou, tak jak je vidět na obrázku č.8. Zde je jasné označené slabé místo v datové vrstvě, kde musí databázový systém provádět takzvaný Full Table Scan, což znamená, že musí sekvenčně procházet všechny záznamy z vybrané tabulky, neboť v podmínce SQL dotazu požadujeme výběr dat ze sloupce tabulky, ke kterému není přiřazený index.

V případě, že by tabulka obsahovala větší množství dat, například několik desítek milionů, cena operace nad touto tabulkou bez indexu bude velmi vysoká. Abychom snížili cenu této operace je zapotřebí zavést pro problémový sloupec index. Zde si ovšem musíme uvědomit, že pokud bychom prováděli tuto operaci nad produkční databází, mohli bychom způsobit výpadek databáze, neboť při provádění operace vytváření indexu, se uzamkne celá tabulka na dobu trvání operace, což by mohlo způsobit kolaps IS. V případě, že tabulka obsahuje velké množství záznamů, může tato operace trvat i několik hodin. Execution plán po přidání indexu můžete vidět na obrázku č.9.

Definice 6.2 *Explain příkaz je základní nástroj pro optimalizaci databáze. Zobrazuje postup databázového systému při vykonávání SQL dotazů a ceny jednotlivých operací.*

```
explain select * FROM 'attendance' 'a' WHERE 'a' . 'users_login' IN ( SELECT DISTINCTROW 'us
    ' . 'Login' FROM 'sheduleitem' 'si' JOIN 'studygroup' 'sg' ON 'si' . 'StudyGroup_idStudyGroup' =
    'sg' . 'idStudyGroup' JOIN 'users' 'us' ON 'us' . 'StudyGroup_idStudyGroup' = 'sg' . '
    idStudyGroup' WHERE 'si' . 'Users_Login' = 'Cager' AND 'sg' . 'SchoolYear_idSchoolYear' = 15
    );
```

Výpis 11: Ukázka příkazu EXPLAIN

Výstupem tohoto příkazu jsou informace o efektivitě vykonání dotazu. Zejména potom informace o indexech, počtu řádků, které bylo nutné projít při vykonání dotazu a další informace. Výsledek je vlastně textová forma výše zmiňovaného příkazu Show execution plán, jehož výstupem je grafické znázornění těchto informací.

```
select * from sys.'x$statement_analysis';
```

Výpis 12: Ukázka MySQL kódu pro vypsání dotazů s vysokou cenou

Wonders WP5
Performance Reports

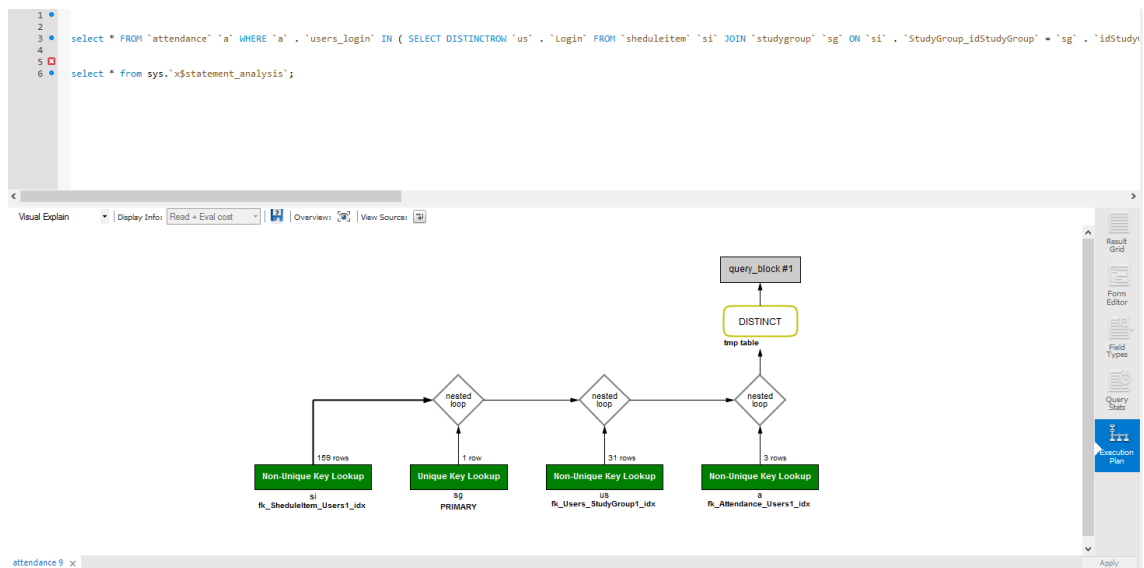
Report
▼ Hot Spots for I/O
Top File I/O Activity Report
Top I/O by File by Time
Top I/O by Event Category
Top I/O by User/Thread
▼ High Cost SQL Statements
Statement Analysis
Statements in Highest 5 Percent by Ru...
Using Temp Tables
With Sorting
Full Table Scans
Errors or Warnings
Database Schema Statistics
Schema Object Overview (High Overh...
Schema Index Statistics
Schema Table Statistics
Tables with Full Table Scans
▼ Wait Times (Top 10)
Global Wait Time
Wait Classes by Time
Wait Classes by Average Time
▼ InnoDB Statistics
InnoDB Buffer Stats by Schema
InnoDB Buffer Stats by Table

Statement Analysis

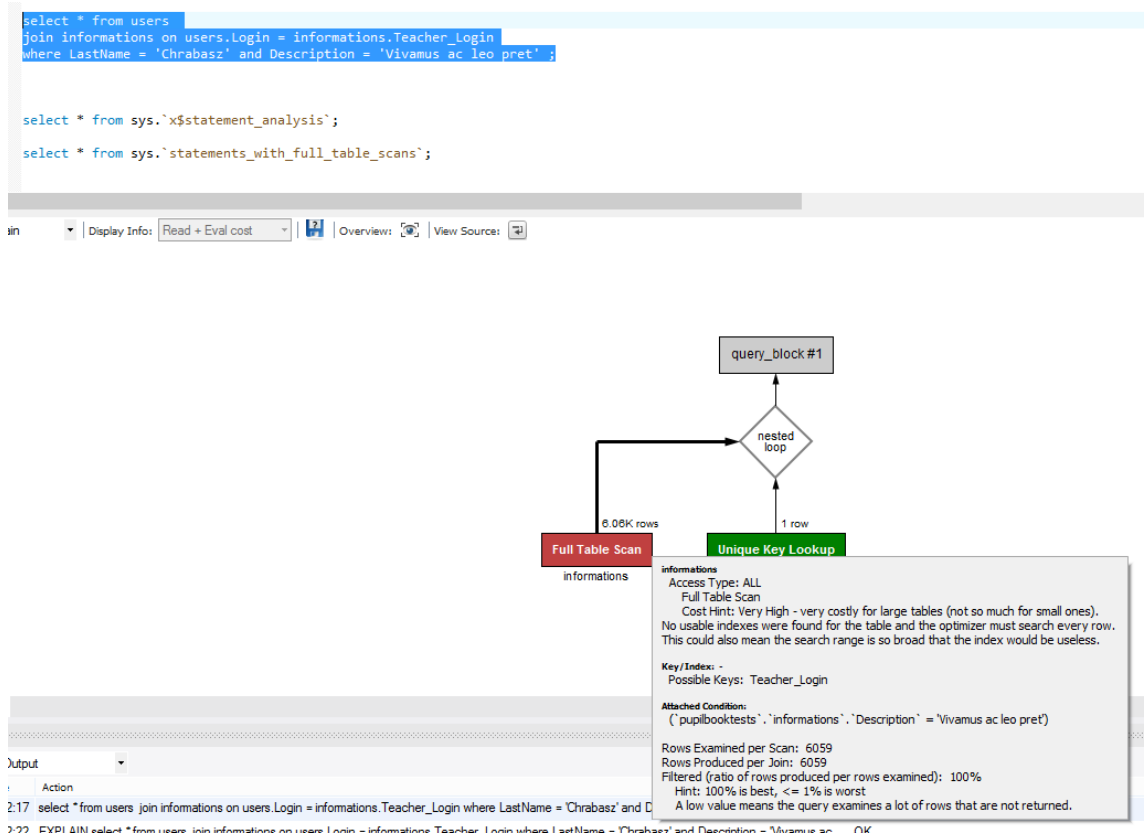
List statements with various aggregated statistics

Query	Full Table S...	Executed (#)	Errors (#)	Warnings (#)	Total Time (s...	Max Time (us)	Avg Time (us)	Rows Sent (#)	Avg. Rows ...	Rows Scann...
SELECT * FROM 'attendances', 's' WHERE 's'. 'users_login' IN (SELECT DISTINCTROW 'us', 'login' FROM 'users_login', 'role', 'birthdate', 'deleted', 'email', 'firstname', 'lastname', 'middlename', 'pas...		1181	0	0	0.119410904...	179326.89	100726.46	649227	5469	18972372
COMMIT		837684	0	0	0.1578233364...	38825.83	661.09	837684	0	837684
SELECT * FROM 'results' WHERE 'Teacher_Login' = ? AND 'SchoolYear_IdSchoolYear' = ?		179615	0	0	0.49982739113	296534.74	2857.81	0	0	0
SELECT DISTINCTROW 's', 'Firstname', 's', ' Middlename', 's', 'Lastname', 's', 'Phone', 's', 'E...		1083	0	0	0.12494819050	27039.34	114818.27	766916	708	4584400
INSERT INTO 'results' ('Description', 'Score', 'DATE', 'Teacher_Login', 'Student_Login', 'StudySubje...		113416	0	0	0.6940024705	28033.83	611.91	0	0	0
SELECT * FROM 'results' WHERE 'Student_Login' = ? AND 'SchoolYear_IdSchoolYear' = ? AND 'StudySubj...		140644	0	0	0.3403566920	98182.25	242.61	57084	4	115204
SELECT @@ SESSION. 'tx_read_only'		148855	0	0	0.322650647	13388.39	216.76	148855	1	0
SHOW GLOBAL STATUS		6698	0	0	0.210691176	14584.11	3149.73	2284018	341	2284018
SET character_set_results = NULL		28973	0	0	0.1837156210	20706.25	634.09	0	0	0
SELECT AVG ('Score') FROM 'results' WHERE 'SchoolYear_IdSchoolYear' FROM 'studygroup' WHERE ('IdStudyGroup' = ...		47950	0	0	0.1750173978	30900.45	365.00	47950	1	47950
SELECT COUNT (' ') FROM 'users' 'u' WHERE 'u'. 'role' = ? AND 'u'. 'password' = ? AND 'Role' = ?		572	0	0	0.1447359373	120335.47	25301.72	1255733	265	2699999
SELECT COUNT (' ') FROM 'users' 'u' WHERE 'u'. 'login' = ? AND 'u'. 'password' = ? AND 'Role' = ?		17963	0	0	0.1284931941	16463.39	636.04	20202	1	20202
INSERT INTO 'attendance' ('MissingStart', 'MissingEnd', 'Excused', 'Users_Login') VALUES (...)		18527	0	0	0.1028813194	76716.96	624.15	17963	1	53885
SET NAMES 'utf8'		35616	0	0	0.32561654	29129.12	491.29	0	0	0
SELECT 'IdStudySubject', 'NAME', 'ShortName' FROM 'studysubject' WHERE ('IdStudySubject' = ?)		27328	0	0	0.53120410	16682.94	182.97	17966	1	17966
SELECT DISTINCTROW 'Description' FROM 'results' JOIN 'users' ON 'results'. 'Student_Login' = 'users' ...		114	0	0	0.53120410	201768.10	45910.00	422	4	30584
INSERT INTO 'users' ('Firstname', 'Lastname', 'Phone', 'Email', 'Login', 'PASSWORD', 'StudyGroup_I...		6318	0	0	0.495120512	67628.35	768.54	0	0	0
SELECT * FROM 'sheduleitem' LEFT JOIN 'studygroup' ON 'sheduleitem'. 'StudyGroup_IdStudyGroup' = 's'.		4830	30	0	0.37105174	78905.81	768.54	61292	54	326456
SELECT DISTINCTROW 'studysubject'. 'Name', 'studysubject'. 'IdStudySubject', 'studysubject'. 'Short...		1184	0	0	0.31201714	24675.92	2645.28	14064	12	142596
SHOW VARIABLES WHERE 'Variable_name' = ? OR 'Variable_name' = ? OR 'Variable_name' = ? OR 'Variabl...		1288	0	0	0.29242214	22286.32	2268.56	30875	24	390720
SELECT DISTINCTROW 's'. 'Firstname', 's'. ' Middlename', 's'. 'Lastname', 's'. 'Phone', 's'. 'E...		1445	0	0	0.25862536	9269.53	1796.57	26010	18	26010
INSERT INTO 'parentstudent' ('SQL_TSLDAY', 'SQL_TSL_HOUR', 'StudyGroup_IdStudyGroup', 'Users_Login' ...		3240	0	0	0.191045694	17334.86	589.65	0	0	0
DELETE FROM 'informations' WHERE 'Teacher_Login' = ? ORDER BY CreateDate DESC		240	0	0	0.17257484	12111.76	717.74	0	0	0
SELECT * FROM 'attendances', 'results' 's' WHERE 's'. 'users_login' = ?		1	0	0	0.107463188	21704.04	4501.89	71879	292	143798
SELECT COUNT (' ') FROM 'users' 'u' WHERE 'u'. 'login' = ? AND 'u'. 'password' = ?		1185	0	0	0.88123187	1048793.06	0	0	0	5814
SELECT * FROM 'results' WHERE 'StudySubject_IdStudySubject' = ? AND 'Teacher_Login' = ? AND 'Student...		1135	0	0	0.56911178	7701.23	501.24	9399	8	9399
SELECT * FROM 'users' 'u' WHERE 'u'. 'role' = ? AND 'deleted' = ?		103	0	0	0.4097142	13918.80	997.89	1359	13	1359
SELECT @@ SESSION. 'auto_increment_increment'		14	0	0	0.18269101	18488.92	1370.79	29059	2075	53892
SELECT * FROM 'users' 'u' WHERE 'u'. 'role' = ? AND 'u'. 'lastname' LIKE ? AND 'deleted' = ?		1445	0	0	0.15044862	1983.32	104.12	1445	1	0
SELECT 'Login', 'Role', 'BirthDate', 'deleted', 'Email', 'Firstname', 'Lastname', ' Middlename', 'PAS...		162	0	0	0.11493170	5613.23	688.23	4680	29	9720
SELECT * FROM 'sys'. 'xstatement_analysis' SELECT 'performance_schema'. 'events_statements_summar...		1	0	0	0.10727356	107273.56	107273.56	291	291	582
DELETE FROM 'studybooktests'. 'users'		133	0	0	0.9477637	2957.15	712.60	263	2	797
SET 'sql_mode' = ?		1	0	0	0.926724	92672.4	92672.4	0	0	270
CREATE DATABASE IF NOT EXISTS 'performance_schema' ENGINE=PERFORMANCE_SCHEMA; SET @@SESSION. 'performance...		1445	0	0	0.8545815	879.79	59.14	0	0	0

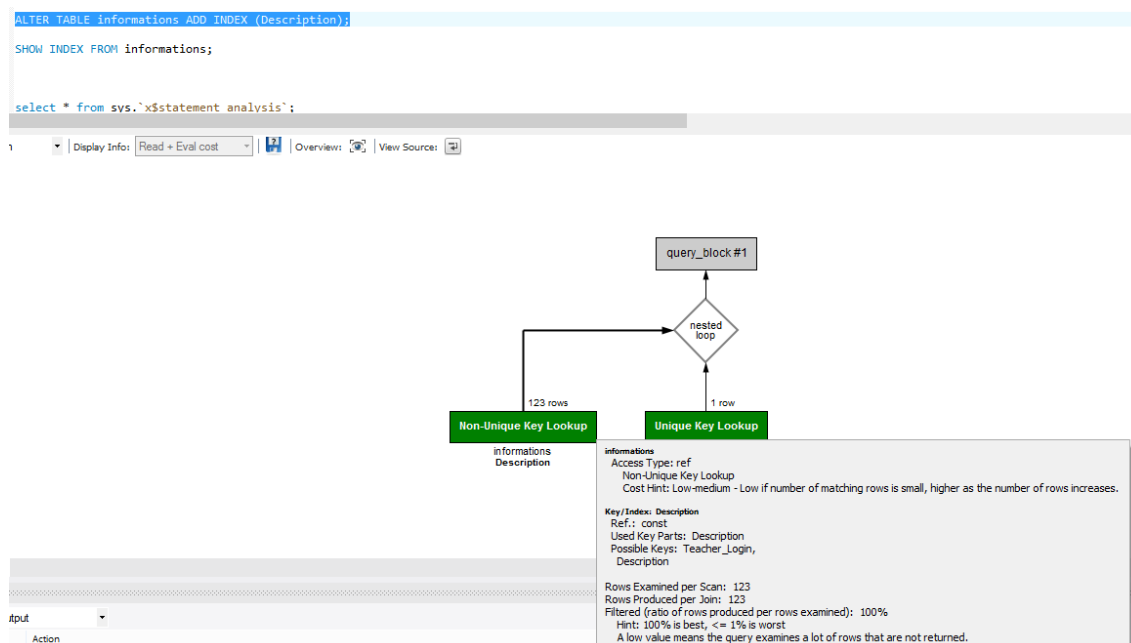
Obrázek 6: Ukázka performance reports MySQL Workbench.



Obrázek 7: Vizuální zobrazení průběhu vykonání dotazu.



Obrázek 8: Vizuální zobrazení průběhu vykonání dotazu, příklad chybného indexu.



Obrázek 9: Vizuální zobrazení průběhu vykonání dotazu, chybějící index doplněn.

```
select * from sys.`statements_with_full_table_scans`;
```

Výpis 13: Ukázka MySQL kódu pro vypisání tabulek nad kterými je často prováděn sekvenční průchod.

6.3 Výsledky testování

Testování jsem prováděl na několika sestavách různého výkonu. Vzhledem k cíli, jímž bylo snížení nákladů na provoz tohoto IS, jsem aplikaci testoval na zařízení Raspberry s procesorem Broadcom BCM2836 ARMv7 Quad Core 900MHz, 1 Gb ram, 8Gb SD kartou Samsung Micro SDHC 8GB Class 10 a zařízení Acer Aspire One s procesorem Intel Atom N270 1.6 Ghz, 1 Gb ram, 16 Gb SSD diskem. Další testovacím strojem byl VPS server s jedním vláknem procesoru Intel Xeon 1.8 GHz, 2 Gb ram a 15 Gb SSD diskem, který mi poskytla firma Wedos a.s. k testování zdarma, a jehož cena je 100 Kč měsíčně bez DPH, což je dle mého názoru velmi nízká cena za takto výkonný virtuální stroj. Nejprve jsem prováděl zátěžové testy, při nichž jsem v rámci generovaných scénářů měnil počet paralelně připojených uživatelů provádějících synchronizaci dat se serverem. Vykonal jsem tři testy s různým počtem uživatelů provádějících synchronizaci mobilní aplikace se serverem.

6.3.1 Prezentace výsledků

Sloupcový graf na obrázku č.10 je rozdělen na 6 podgrafů, na nichž jsou zobrazeny informace o jednotlivých voláních webových služeb. Každý podgraf je složen ze šesti sloupců, kde výška červeného sloupce reprezentuje průměrný čas odezvy volání webové služby, modrý sloupec medián, šedý sloupec minimální čas odezvy a černý pak maximální čas odezvy. Zelený, žlutý a fialový sloupec představují výsledky odezvy line 90%, line 95% a line 99%.

Pravidlo line X%, je největší čas odezvy v případě, že seřadíme všechny časy odezvy podle velikosti a z této řady vybereme právě tu odezvu, která se nachází na pozici X% z celkového počtu měření. Pro příklad tedy mějme měření, v němž bychom naměřili tyto hodnoty 8,9,3,7,5,6,4,1,2,10 v milisekundách. Pokud bychom zde zavedli, pravidlo line 90%, výsledkem by byla hodnota 9 milisekund, neboť po seřazení bychom získali řadu 1,2,3,4,5,6,7,8,9,10 a na pozici 90% z 10 je právě hodnota 9.

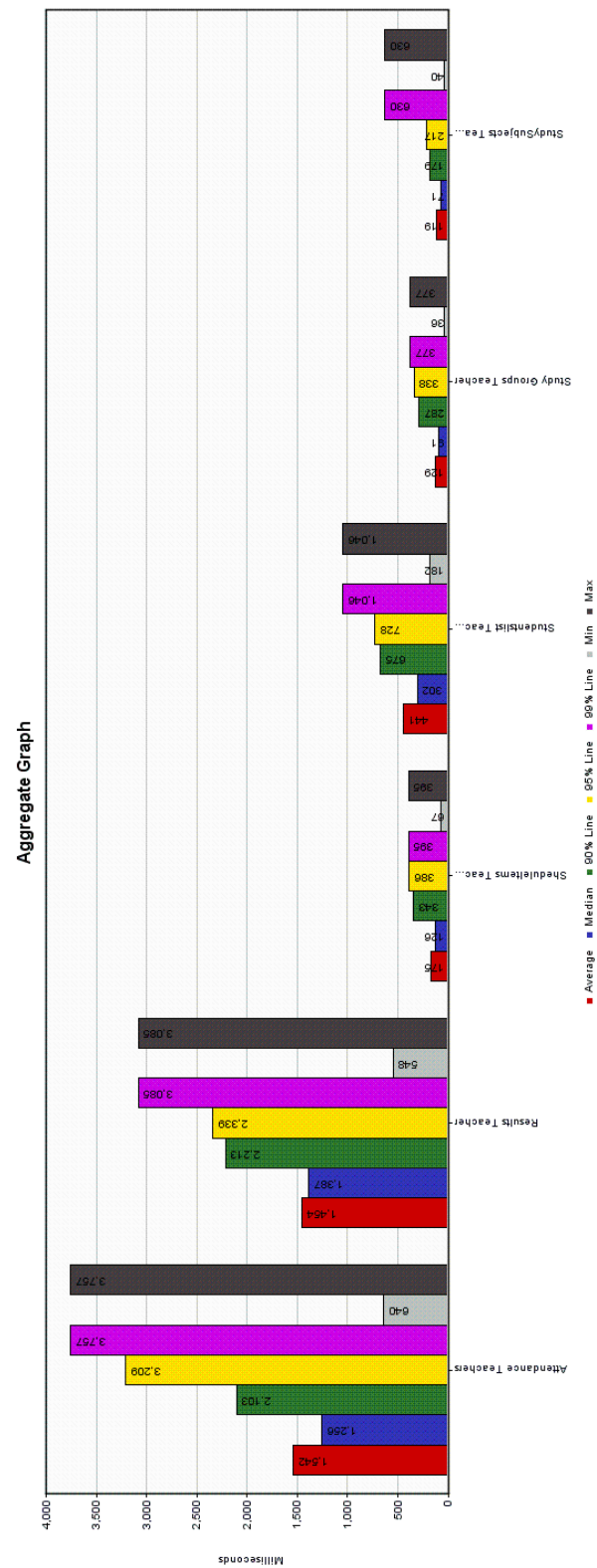
Lineární graf na obrázku č.11 znázorňuje průměrné hodnoty odezvy webových služeb v rámci uvedeného časového intervalu. Všechny grafy lze nalézt v příloze B, C - Testování servisní vrstvy.

6.3.2 Testování na VPS serveru

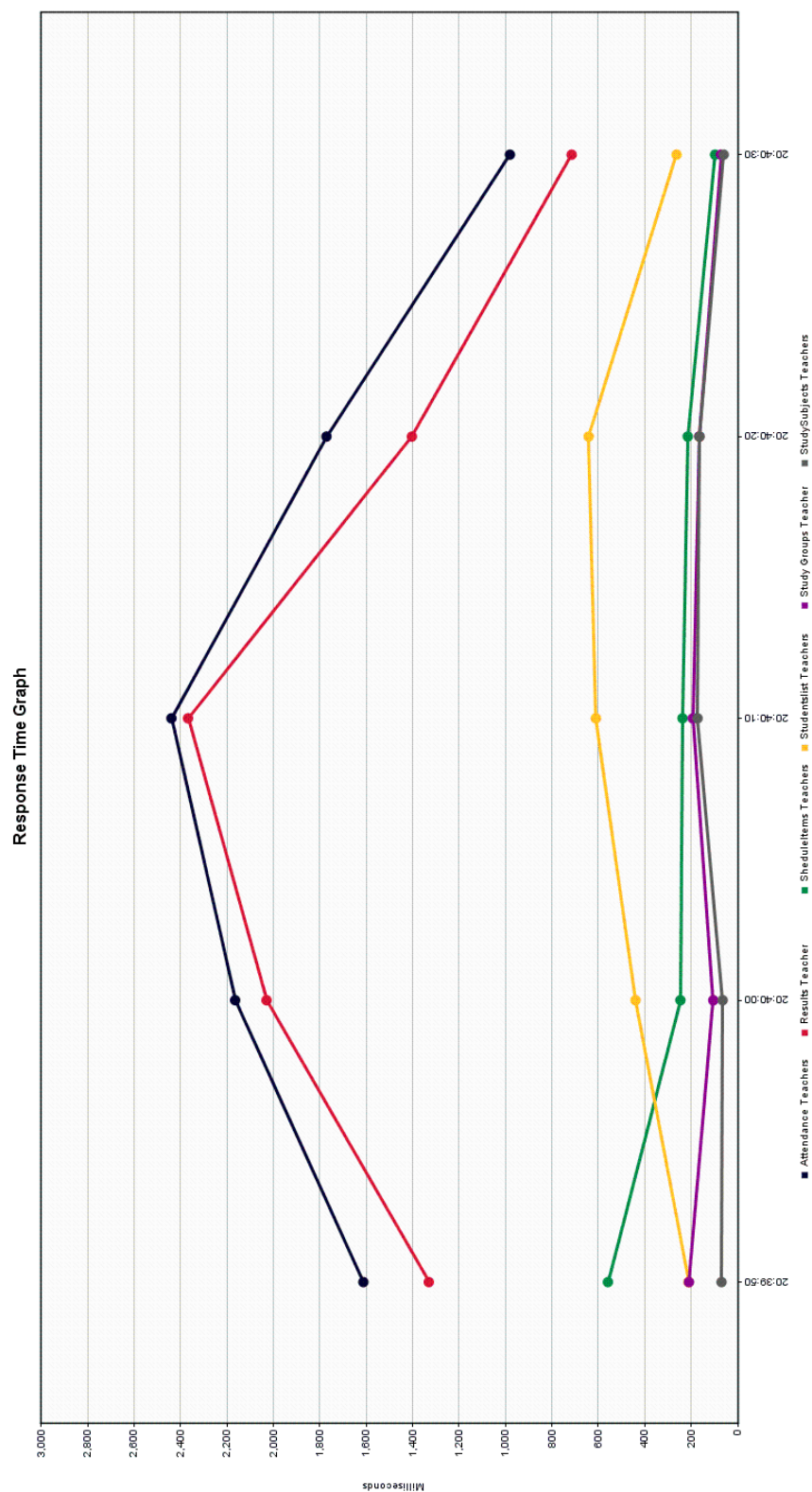
Testování probíhalo vzdáleně, jelikož nebylo možné spustit nástroj Jmeter v serverovně firmy, proto jsem musel zohlednit degradaci výkonu, která vznikla mezi testovaným a testovacím strojem. I když testování probíhalo vzdáleně, odezva webových služeb byla velmi slušná, jak se můžeme přesvědčit v příloze B. Při středním zatížení tedy při paralelním přístupu pěti učitelů a padesáti žáků se čas odezvy pohyboval u žáků okolo jedné sekundy na jedno volání webové služby a u učitele byl čas odezvy průměrně 2 sekundy. Vzhledem k faktu, že mobilní aplikace a webová aplikace pro žáky fungují offline, je takováto odezva velmi dobrá, jelikož bude synchronizace probíhat pouze při spuštění aplikace. Dalším pozitivem je ukládání dat VPS serveru na RAID uložisti spravovaném firmou Wedos, což znamená, že by při ostrém nasazení bylo vyřešeno i zálohování dat. Tento typ nasazení aplikace bych preferoval při ostrém nasazení IS.

6.3.3 Testování Raspberry Pi 2

Překvapením byly výsledky testování na zařízení Raspberry Pi 2. Přestože se jedná o zařízení menší než 2.5 palcový pevný disk a spotřebou 1500mW při 5 voltech, toto zařízení dosahovalo lepší odezvy než níže testovaný Acer Aspire One, jak se můžete přesvědčit v příloze C. Při středním zatížení, tedy při paralelním přístupu pěti učitelů a padesáti žáků, se čas odezvy pohyboval u žáků okolo 2,5 sekundy na jedno volání webové služby a u učitele byl čas odezvy průměrně 4 sekundy. Bohužel na rozdíl od předchozího VPS serveru se nelze spolehnout na datové uložisti, které v tomto případě tvoří jediná SD karta, tudíž není možné na zařízení uchovávat redundantní záznamy z datové vrstvy a pro 100% spolehlivost bych musel řešit ukládání redundantních dat na dalším zařízení. Byť byla odezva na tomto zařízení horší v porovnání s výše zmíněným VPS serverem,



Obrázek 10: Agregční graf učitelé - VPS server, synchronizace 2 učitelů a 20 žáků současně



Obrázek 11: Graf průměrná odezva učitelé - VPS server, synchronizace 2 učitelů a 20 žáků současně

v budoucnu bychom mohli uvažovat provoz IS na nových verzích produktů Raspberry, kde budou jistě výkonnější procesory a větší množství paměti. Pokud bychom uvažovali stejný nárůst výkonu, jako tomu bylo v případě RaspberryPI a RaspberryPI 2, potom bychom mohli dosáhnout podobných výsledků testování, jako tomu bylo v případě VPS serveru společnosti Wedos a mělo by smysl zabývat se zálohováním dat na jiném zařízení.

6.3.4 Testování Acer Aspire One

Ačkoliv testování probíhalo v lokální síti, odezva tohoto stroje se mnohdy přibližovala 10 sekundám, což je doloženo v příloze C, a tudíž nemělo smysl dále uvažovat provoz na tomto zařízení.

7 Závěr

Cílem této bakalářské práce bylo vytvořit IS, takzvanou elektronickou žákovskou knížku a porovnat vytvořený IS se zavedenými systémy. Tento IS má řadu výhod oproti zavedeným systémům. Především pořizovací cena a provozní náklady IS jsou mnohem nižší v porovnání s podobnými systémy, jelikož IS není propojen s žádnými dalšími komerčními produkty a může být provozován na VPS Serveru za 100 Kč měsíčně, který poskytuje dostatečný výkon. Další výhodou je možnost exportu mobilní aplikace pro žáky a rodiče na všechny běžně používané mobilní platformy, což výše zmíněné systémy neumožňují. Nesmíme opomenout, že je IS vyvíjen pod licencí GPL, to znamená, že pořízení IS bude zdarma a do budoucna se můžeme dočkat podpory ze strany mnoha dalších vývojářů, jako tomu bylo u projektu Moodle.

Během implementace IS jsem se neustále snažil optimalizovat kód, aby měla aplikace co nejlepší odezvu a aby byla uživatelsky přívětivá. Při implementaci multiplatformní mobilní aplikace, jsem objevil inovativní framework AngularJS od společnosti Google, díky němuž lze jednoduše vytvářet single page offline aplikace. Je škoda, že jsem tento framework neobjevil dříve, neboť bych na něm rád postavil celý IS. Přesto jsem rád, že jsem se naučil pracovat z frameworky JSF a Primefaces, protože v těchto frameworkcích, lze rychle a levně naimplementovat kvalitní IS.

Za největší pozitivum považuji zájem ze strany škol o tento IS a věřím, že jednoho dne bude školami využíván.

8 Reference

- [1] ŠARMANOVÁ, Jana. *Databázové a informační systémy*, Ostrava: VŠB – Technická univerzita Ostrava, 2007. ISBN 978-80-248-1499-5
- [2] KRÁTKÝ, Michal, BAČA, Radim. *Databázové systémy*, Ostrava: VŠB – Technická univerzita Ostrava, 2010. 368 s.
- [3] JEŽEK, David *Java Technologie*, Ostrava: VŠB – Technická univerzita Ostrava, 2014. 445 s.
- [4] BRADFORD, Ronald. *Effective MySQL: Optimizing SQL Statements*, McGraw-Hill Companies, Inc., 2012. ISBN 978-0071782791
- [5] FOWLER, Martin. *Patterns of Enterprise Application Architecture*, Person Education, Inc., 2003. ISBN 007-6092019909
- [6] MEIER, Reto. *Profesional Android 4 Application Development*, John Wiley & Sons, Inc., 2012. ISBN 978-1118102275
- [7] POWERS, Shelley. *Adding Ajax*, O'Reilly Media, Inc., 2007. ISBN 978-0596529369
- [8] SHOTTES, William E., Jr. *The Linux Command Line*, A LinuxCommand.org Book. ISBN 978-1593273897
- [9] EMS generátor dat pro MySQL web výrobce. URL:<<http://www.sqlmanager.net/en/products/mysql/datagenerator>> [cit. 2015-02-12].
- [10] Datanamic generátor dat pro MySQL web výrobce. URL: <<http://www.datanamic.com/datagenerator-for-mysql/index.html>> [cit. 2015-02-12].
- [11] Generate Data online nástroj URL: <www.generatedata.com> [cit. 2015-02-15].
- [12] JSF - JavaServer Faces URL:<<http://www.oracle.com/technetwork/java/javasee/javaserverfaces-139869.html>> [cit. 2015-02-17].
- [13] PrimeFaces frontend framework URL: <<http://www.primefaces.org/>> [cit. 2015-02-01].
- [14] EclipseLink URL: <<http://eclipse.org/eclipselink/>> [cit. 2015-03-15].
- [15] ORM - Objektově Relační Mapování URL: <http://cs.wikipedia.org/wiki/Objektov%C4%9B_rela%C4%8Dn%C3%AD_mapov%C3%A1n%C3%AD> [cit. 2015-02-16].
- [16] Java API for RESTful Services. URL: <<https://jax-rs-spec.java.net/>> [cit. 2015-03-17].
- [17] REST - Representational State Transfer. URL: <http://en.wikipedia.org/wiki/Representational_state_transfer> [cit. 2015-03-04].

-
- [18] AJAX - Asynchronous JavaScript and XML. URL: <<http://api.jquery.com/jquery.ajax/>> [cit. 2015-04-02].
- [19] JPA - Java Persistence API. URL: <<http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html>> [cit. 2015-02-03].
- [20] Glassfish - Open source Aplikační server. URL: <<https://glassfish.java.net/>> [cit. 2015-02-18].
- [21] JDBC - The Java Database Connectivity. URL: <<http://www.oracle.com/technetwork/java/javase/jdbc/index.html>> [cit. 2015-02-18].
- [22] MySQL Server. URL: <<https://www.mysql.com/>> [cit. 2015-02-05].
- [23] JavaScript. URL: <<http://www.javascript.cz/>> [cit. 2015-03-06].
- [24] GNU General Public License. URL: <<https://www.gnu.org/copyleft/gpl.html>> [cit. 2015-02-11].
- [25] HTML5. URL: <<http://www.html5.cz/>> [cit. 2015-02-19].
- [26] AngularJS. URL: <<https://angularjs.org/>> [cit. 2015-02-18].
- [27] Twitter Bootstrap. URL: <<http://getbootstrap.com/>> [cit. 2015-02-17].
- [28] Android. URL: <<https://www.android.com/>> [cit. 2015-03-17].
- [29] Ubuntu. URL: <<http://www.ubuntu.cz/>> [cit. 2015-03-18].
- [30] Testování výkonu webových aplikací. URL: <<http://vsadnajavu.cz/2010-02/nezarazene/testovani-vykonu-webovych-aplikaci/>> [cit. 2015-03-17].
- [31] Edookit - Moderní systém pro moderní školu. URL: <<http://www.edookit.cz/>> [cit. 2015-04-11].
- [32] Bakaláři - Program pro školní administrativu. URL: <<http://www.bakalari.cz/uvod.aspx>> [cit. 2015-04-11].
- [33] Všetko pre školu - Program pro školní administrativu. URL: <<http://www.vsetkopreskolu.sk/>> [cit. 2015-04-11].

A Datový slovník

<i>Atribut</i>	<i>Dattyp</i>	<i>Velikost</i>	<i>PK</i>	<i>Null</i>	<i>Index</i>	Popis atributu
idSchoolYear	INT		ANO	NE	ANO	Primární klíč školního roku
name	VARCHAR	20	NE	NE	NE	Název školního roku
isActualYear	BOOLEAN		NE	NE	NE	Určuje, zda je vybraný školní rok právě ten aktuální
StartDate	DATE		NE	ANO	NE	Začátek školního roku
EndDate	DATE		NE	ANO	NE	Konec školního roku

Tabulka 1: Datový slovník - tabulka SchoolYear

<i>Atribut</i>	<i>Dat.Typ</i>	<i>Velikost</i>	<i>PK</i>	<i>Null</i>	<i>Index</i>	Popis atributu
idStudyGroup	INT		ANO	NE	ANO	Primární klíč školního roku
name	VARCHAR	45	NE	NE	NE	Název studijní skupiny
SchoolYear idSchoolYear	INT		NE	NE	ANO	Cizí klíč - Školní rok

Tabulka 2: Datový slovník - tabulka Studygroup

<i>Atribut</i>	<i>Dat.Typ</i>	<i>Velikost</i>	<i>PK</i>	<i>Null</i>	<i>Index</i>	Popis atributu
FirstName	VARCHAR	45	NE	NE	NE	Jméno
MiddleName	VARCHAR	45	NE	ANO	NE	
LastName	VARCHAR	45	NE	NE	ANO	Příjmení
Phone	VARCHAR	15	NE	NE	NE	Telefon
Email	VARCHAR	100	NE	ANO	NE	Email
Login	VARCHAR	255	ANO	NE	ANO	Přihlašovací jméno
Password	VARCHAR	255	NE	NE	NE	Přihlašovací heslo
BirthDate	DATE		NE	NE	NE	Datum narození
studygroup idstudygroup	INT		NE	ANO	NE	Cizí klíč na studijní skupinu
Role	CHAR	1	NE	NE	NE	Role v IS
deleted	BOOLEAN		NE	NE	NE	Určí, zda byl uživatel smazán
SchoolYear idSchoolYear	INT		NE	NE	ANO	Cizí klíč - Školní rok

Tabulka 3: Datový slovník - tabulka Users

<i>Atribut</i>	<i>Dat.Typ</i>	<i>Velikost</i>	<i>PK</i>	<i>Null</i>	<i>Index</i>	Popis atributu
idInformationsp	INT		ANO	NE	ANO	Primární klíč informací
Description	VARCHAR	100	NE	NE	NE	Popis informace
InfoForParrents	BOOLEAN		NE	NE	NE	Určí, zda je informace určena pouze rodičům
SomeMessage	VARCHAR	2000	NE	NE	NE	Samotná informace
CreateDate	DATE		NE	NE	NE	Datum vytvoření informace
StudyGroup idStudyGroup	INT		NE	NE	ANO	Cizí klíč na studijní skupinu
Users Login	INT		NE	NE	ANO	Cizí klíč na uživatele, kterému je informace určena
Teacher Login	INT		NE	NE	ANO	Cizí klíč na uživatele, kterému je informace určena

Tabulka 4: Datový slovník - tabulka Informations

<i>Atribut</i>	<i>Dat.Typ</i>	<i>Velikost</i>	<i>PK</i>	<i>Null</i>	<i>Index</i>	Popis atributu
idSheduleItem	INT		ANO	NE	ANO	Primární klíč položky rozvrhu hodin
Day	TINYINT		NE	ANO	NE	Den
Hour	TINYINT		NE	ANO	NE	Hodina
StudyGroup idStudyGroup	INT		NE	NE	ANO	Cizí klíč na studijní skupinu
Users Login	INT		NE	ANO	ANO	Cizí klíč na učitele, který vyučuje vybraný předmět
StudySubject idStudySubject	INT		NE	ANO	ANO	Cizí klíč na předmět, který se ve vybranou hodinu vyučuje

Tabulka 5: Datový slovník - tabulka Sheduleitem

<i>Atribut</i>	<i>Dat.Typ</i>	<i>Velikost</i>	<i>PK</i>	<i>Null</i>	<i>Index</i>	Popis atributu
idResults	INT		ANO	NE	ANO	Primární klíč klasifikace
Description	VARCHAR	255	NE	NE	NE	Informace o klasifikaci
Score	TINYINT		NE	NE	NE	Hodnocení
Date	DATETIME		NE	NE	NE	Datum vytvoření klasifikace
StudySubject idStudySubject	INT		NE	NE	ANO	Cizí klíč na studijní předmět ze kterého byl žák klasifikován
Users Login	INT		NE	NE	ANO	Cizí klíč na uživatele, který byl klasifikován
Teacher Login	INT		NE	NE	ANO	Cizí klíč na učitele, který klasifikoval žák
SchoolYear idSchoolYear	INT		NE	NE	ANO	Cizí klíč na školní rok ve kterém byl žák klasifikován

Tabulka 6: Datový slovník - tabulka Results

<i>Atribut</i>	<i>Dat.Typ</i>	<i>Velikost</i>	<i>PK</i>	<i>Null</i>	<i>Index</i>	Popis atributu
idAttendance	INT		ANO	NE	ANO	Primární klíč docházka.
MissingStart	DATE		NE	NE	NE	Začátek absence.
MissingEnd	DATE		NE	ANO	NE	Konec absence
Excused	BOOLEAN		NE	NE	NE	Určí, zda je absence omluvena.
Users Login	INT		NE	NE	ANO	Cizí klíč na žáka.

Tabulka 7: Datový slovník - tabulka Attendance

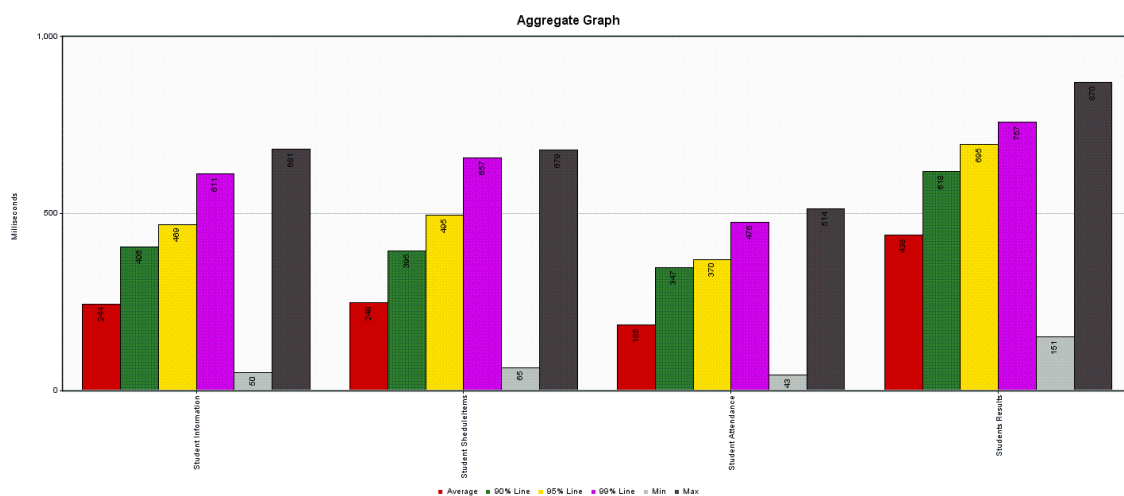
<i>Atribut</i>	<i>Dat.Typ</i>	<i>Velikost</i>	<i>PK</i>	<i>Null</i>	<i>Index</i>	Popis atributu
idParrentStudent	INT		ANO	NE	ANO	Primární klíč záznamu.
Student Login	VARCHAR	255	NE	NE	ANO	Cizí klíč na žáka
Parrent Login	VARCHAR	255	NE	NE	ANO	Cizí klíč na rodiče

Tabulka 8: Datový slovník - tabulka ParrentStudent

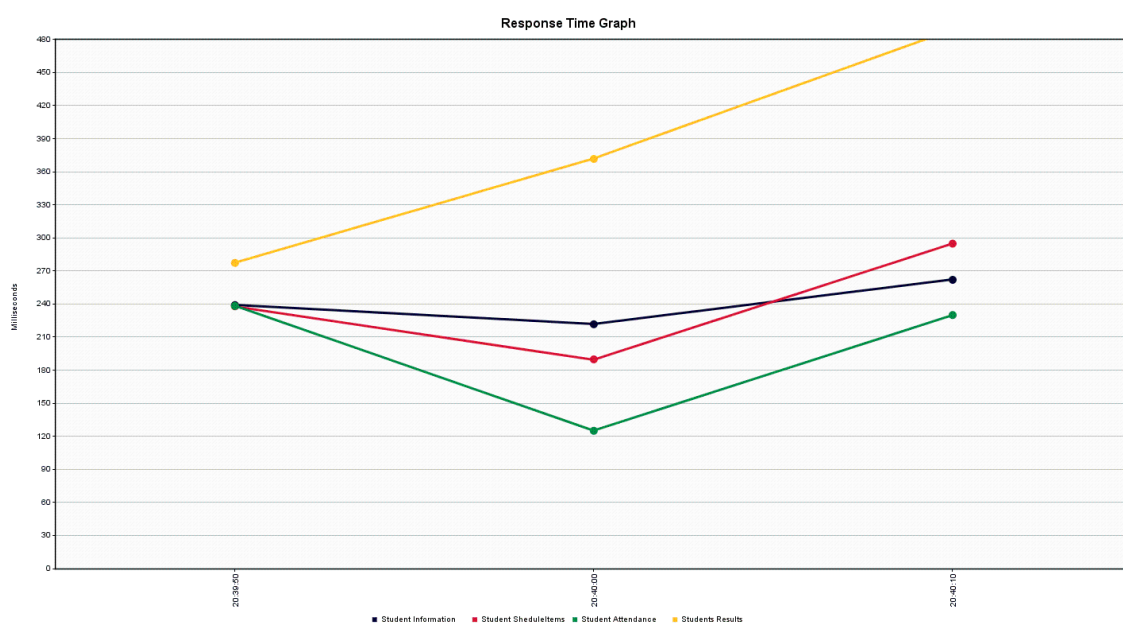
<i>Atribut</i>	<i>Dat.Typ</i>	<i>Velikost</i>	<i>PK</i>	<i>Null</i>	<i>Index</i>	Popis atributu
idStudySubject	INT		ANO	NE	ANO	Primární klíč předmětu.
Name	VARCHAR	45	NE	NE	NE	Název předmětu
Short Name	VARCHAR	5	NE	NE	NE	Zkratka názvu předmětu

Tabulka 9: Datový slovník - tabulka Studysubject

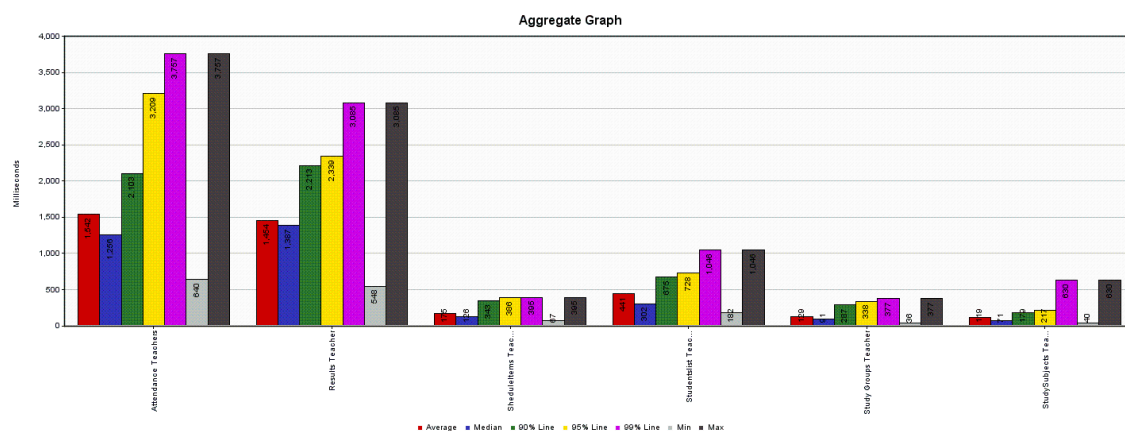
B Testování servisní vrstvy - Virtuální server Wedos



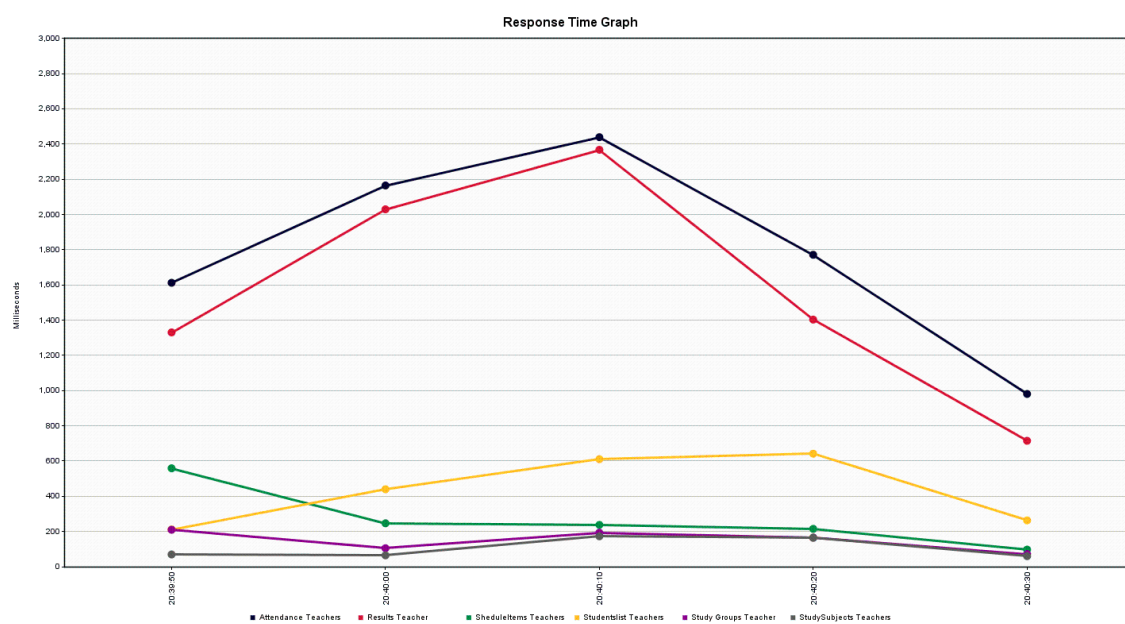
Obrázek 12: Agregační graf žáci - VPS server, synchronizace 2 učitelů a 20 žáků současně



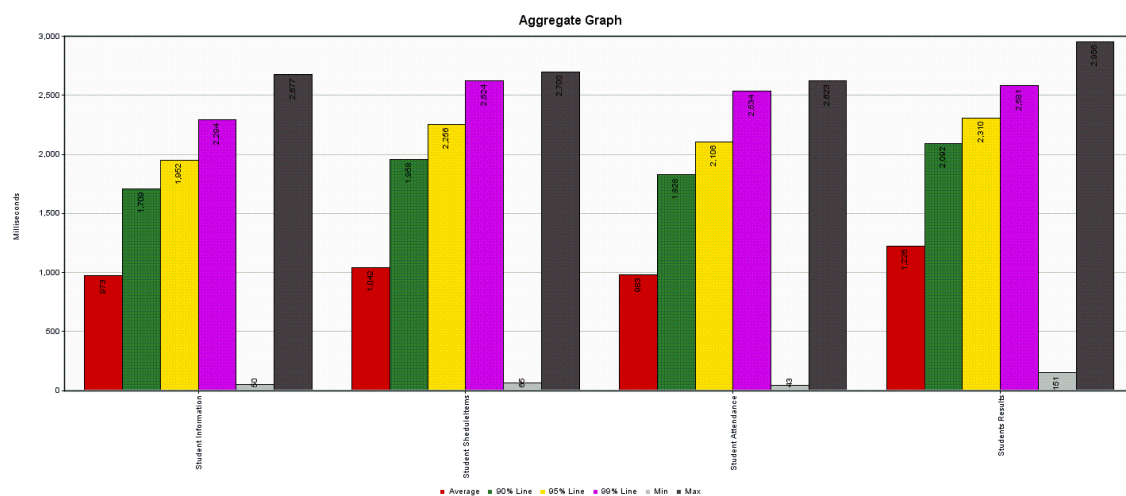
Obrázek 13: Graf průměrná odezva žáci - VPS server, synchronizace 2 učitelů a 20 žáků současně



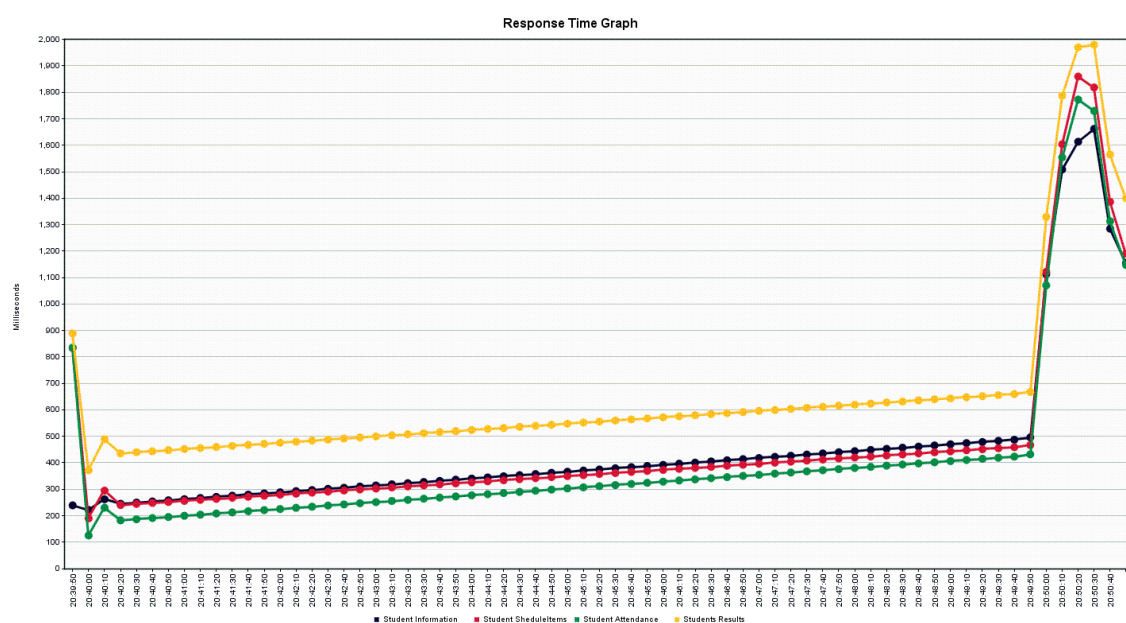
Obrázek 14: Agregací graf učitelé - VPS server, synchronizace 2 učitelů a 20 žáků současně

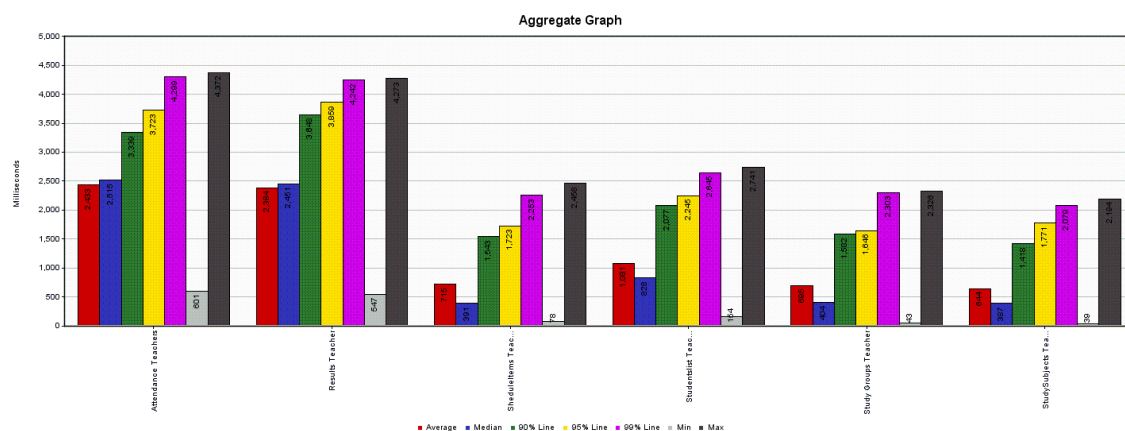


Obrázek 15: Graf průměrná odezva učitelé - VPS server, synchronizace 2 učitelů a 20 žáků současně

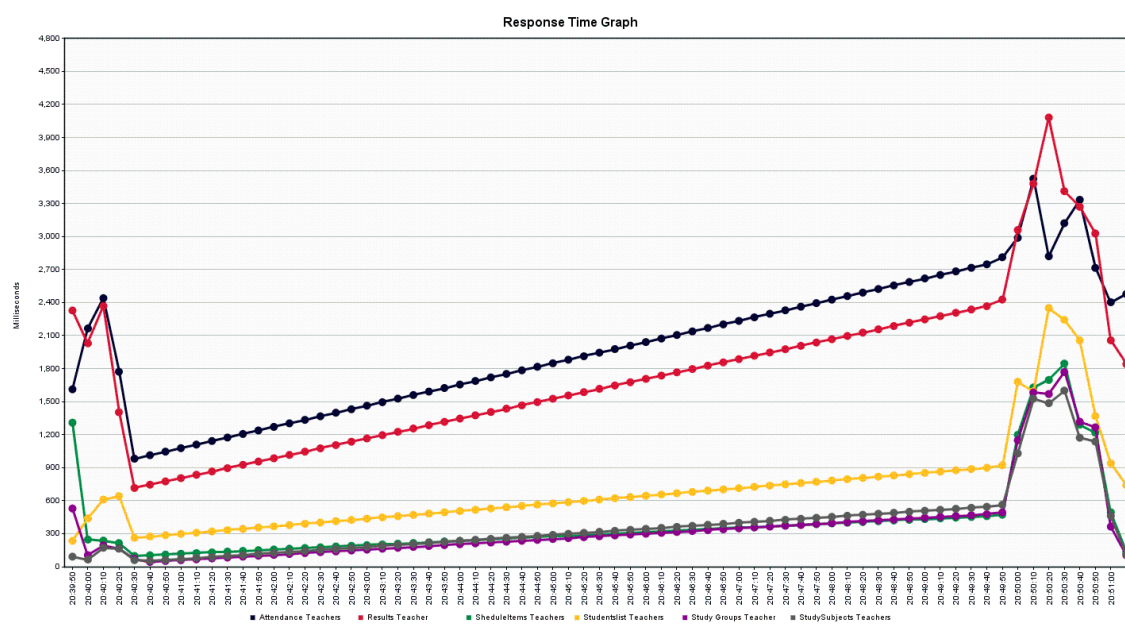


Obrázek 16: Agregační graf žáci - VPS server, synchronizace 5 učitelů a 50 žáků současně

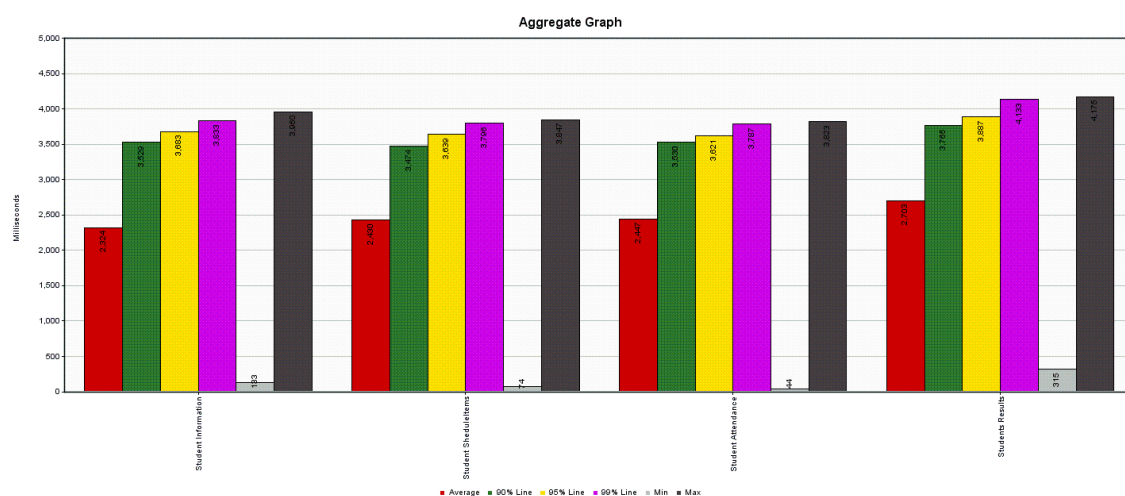




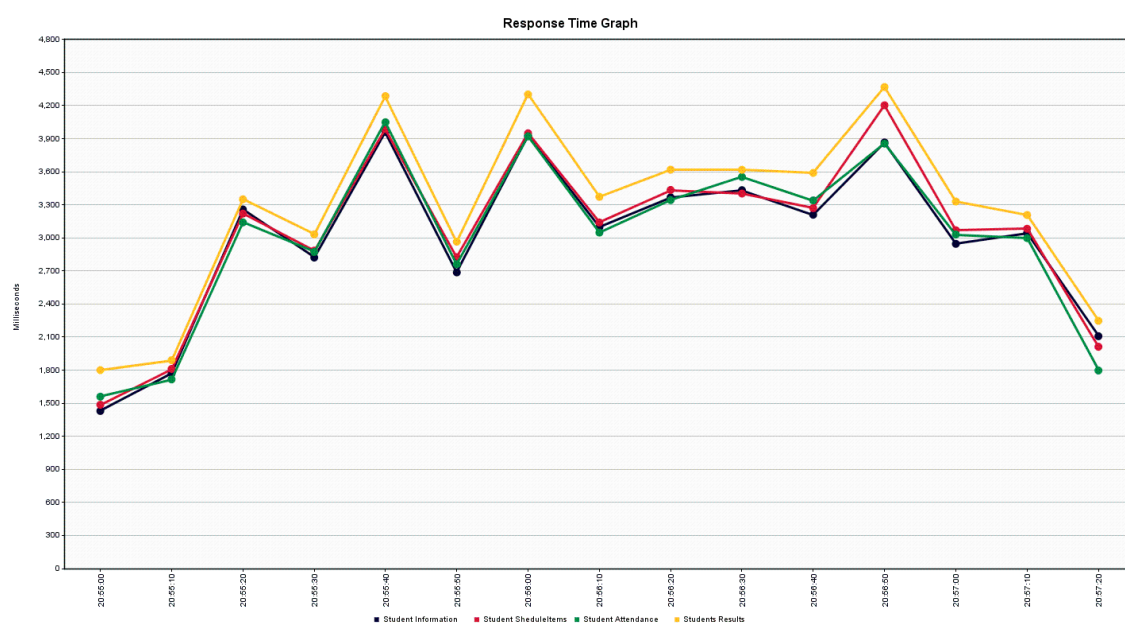
Obrázek 18: Agregací graf učitelé - VPS server, synchronizace 5 učitelů a 50 žáků současně



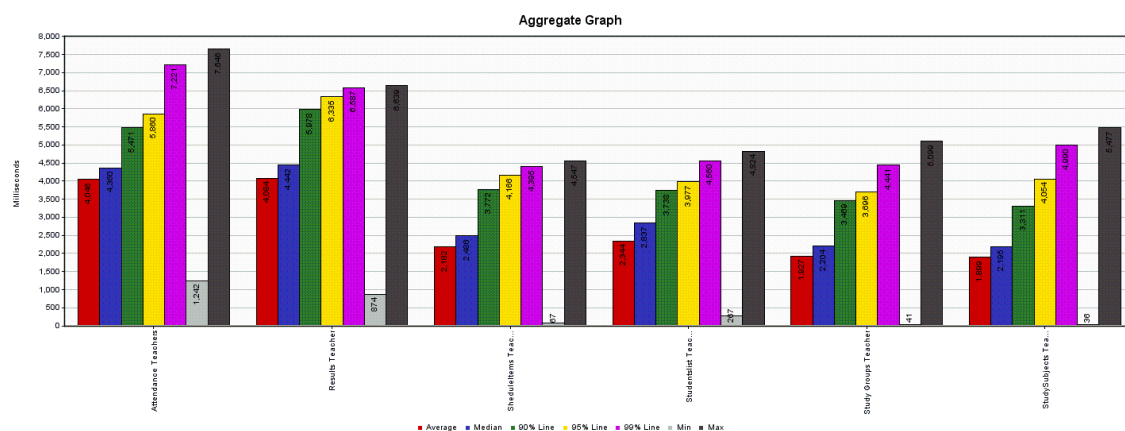
Obrázek 19: Graf průměrná odezva učitelé - VPS server, synchronizace 5 učitelů a 50 žáků současně



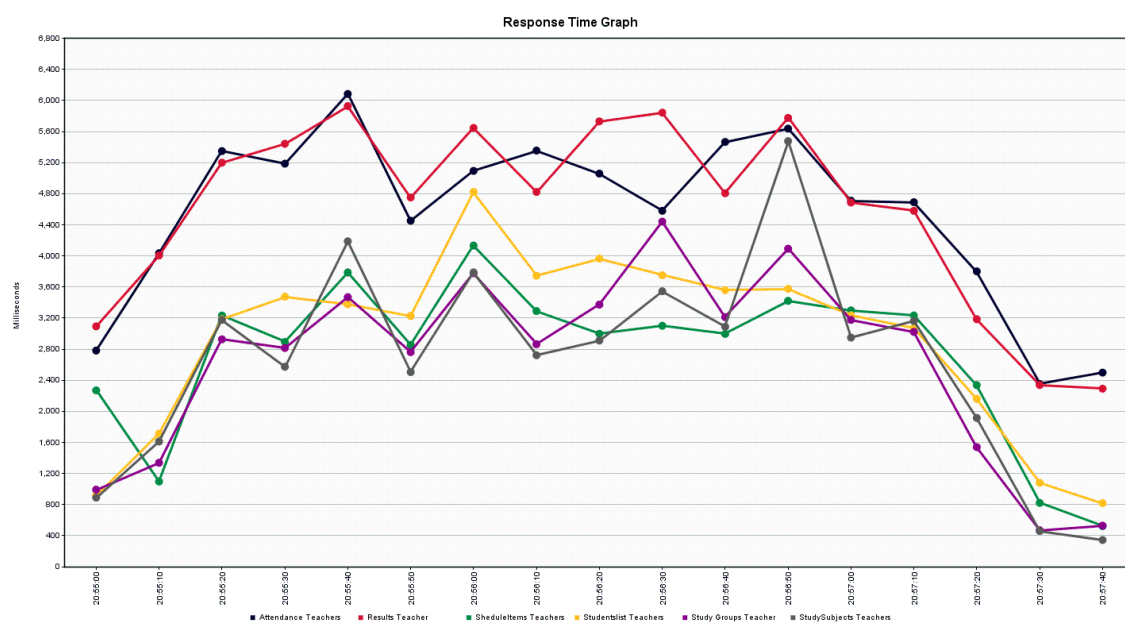
Obrázek 20: Agregací graf žáci - VPS server, synchronizace 10 učitelů a 100 žáků současně



Obrázek 21: Graf průměrná odezva žáci - VPS server, synchronizace 10 učitelů a 100 žáků současně

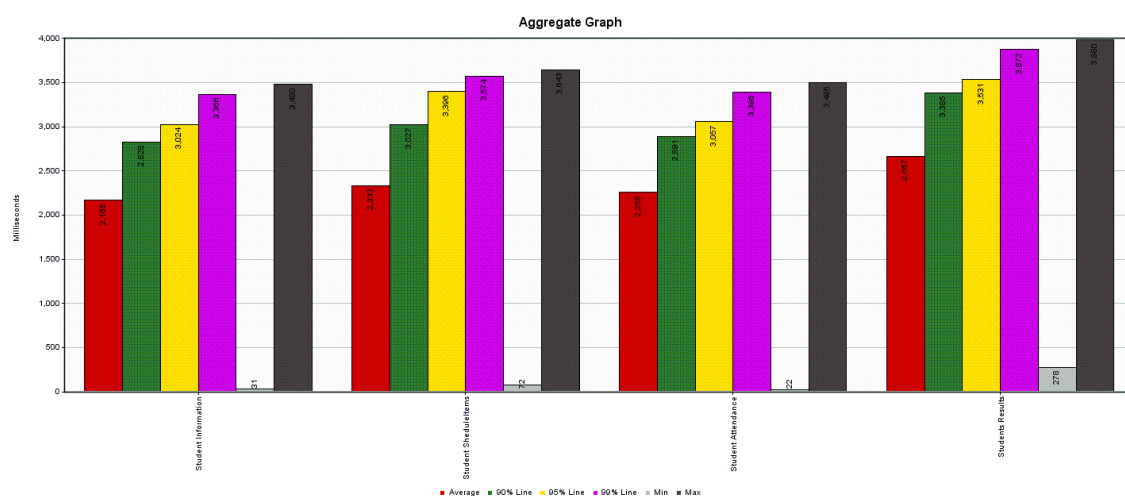


Obrázek 22: Agregací graf učitelé - VPS server, synchronizace 10 učitelů a 100 žáků současně

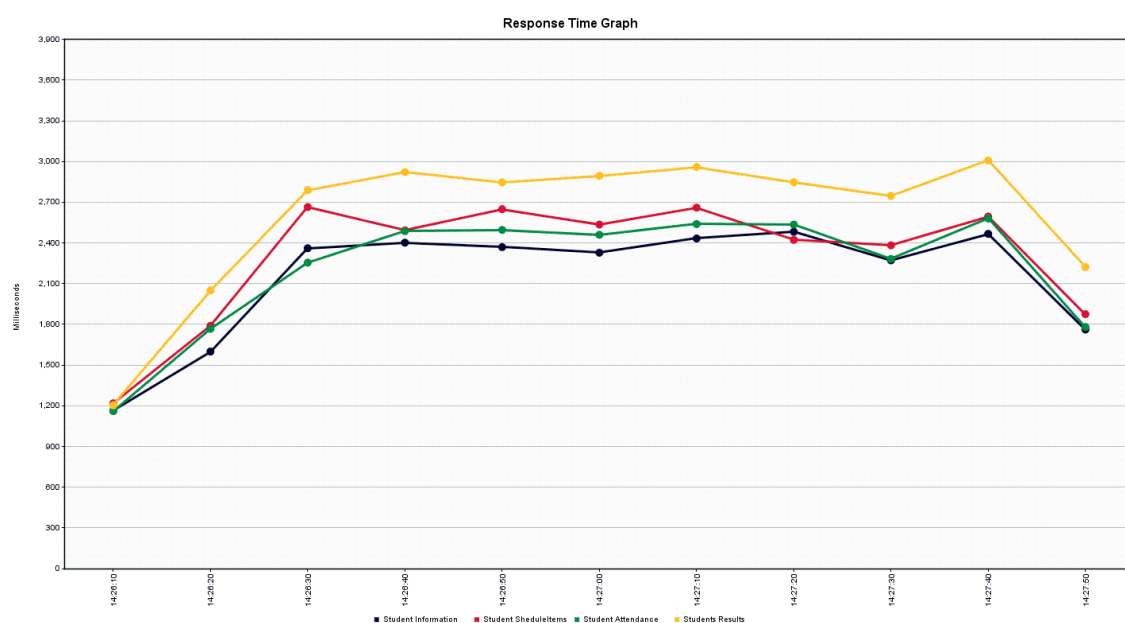


Obrázek 23: Graf průměrná odezva učitelé - VPS server, synchronizace 10 učitelů a 100 žáků současně

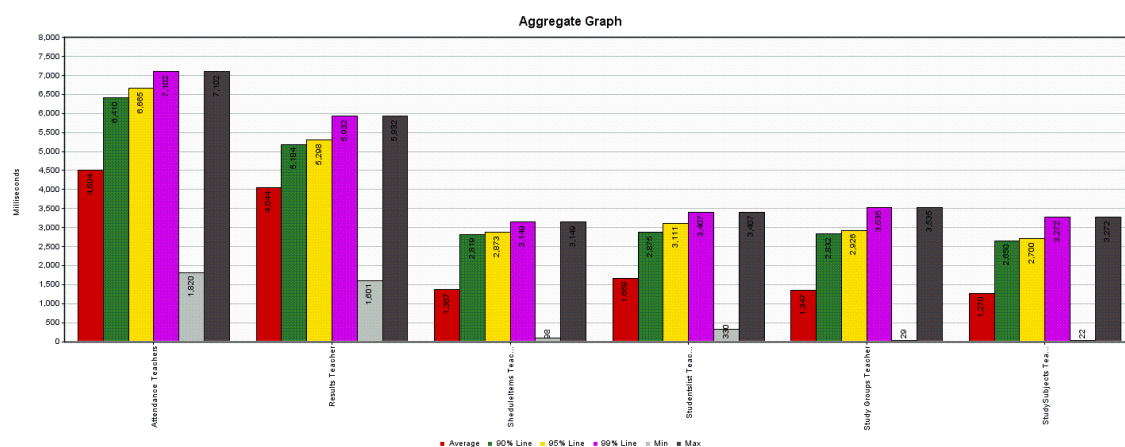
C Testování servisní vrstvy - Raspberry Pi, Aspire One Net-Book



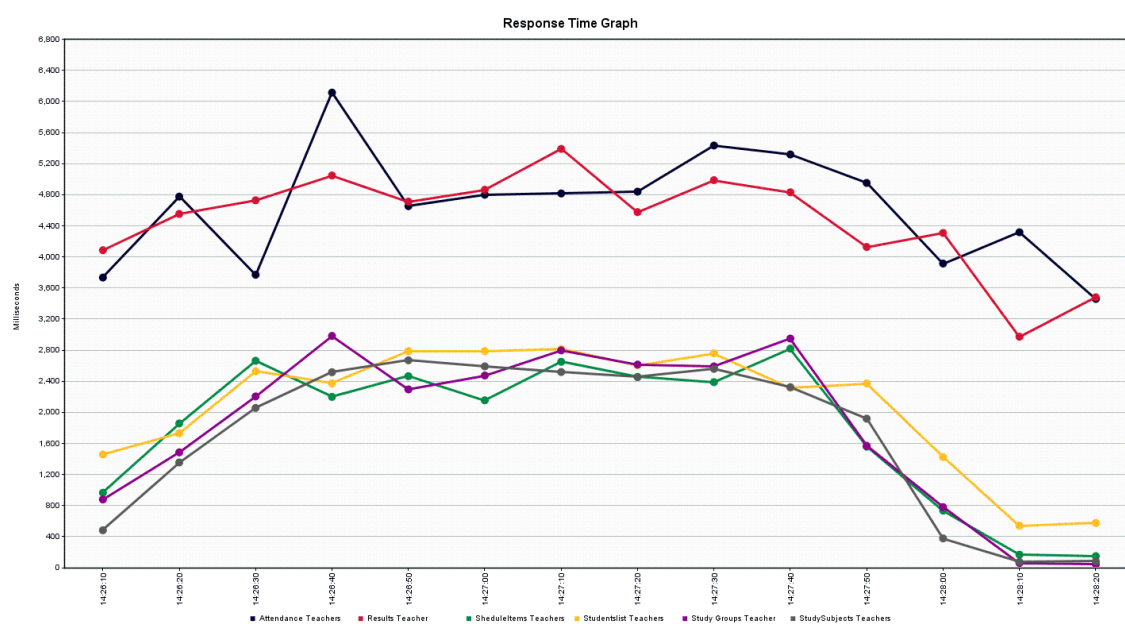
Obrázek 24: Agregací graf žáci - RaspberryPI, synchronizace 5 učitelů a 50 žáků současně



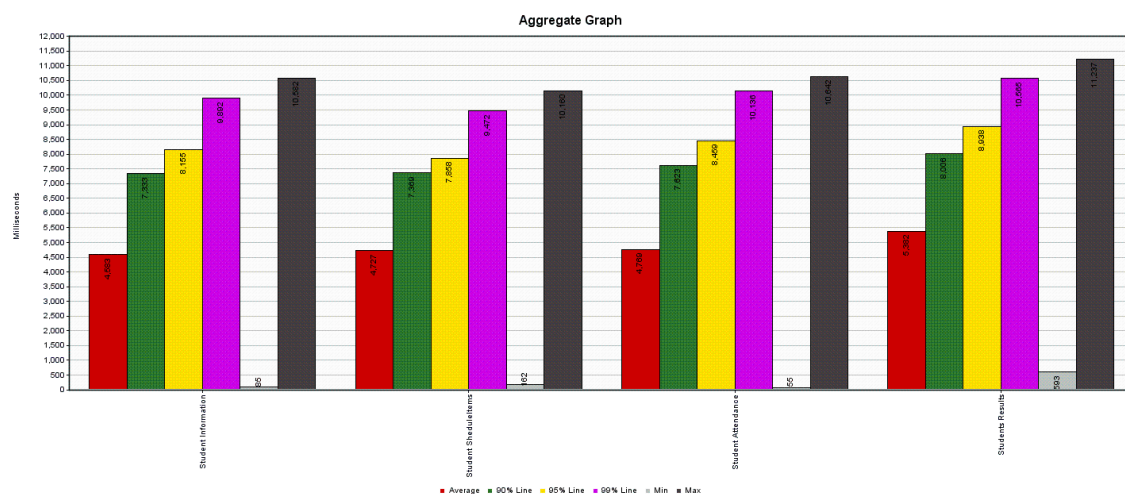
Obrázek 25: Graf průměrná odezva žáci - RaspberryPI, synchronizace 5 učitelů a 50 žáků současně



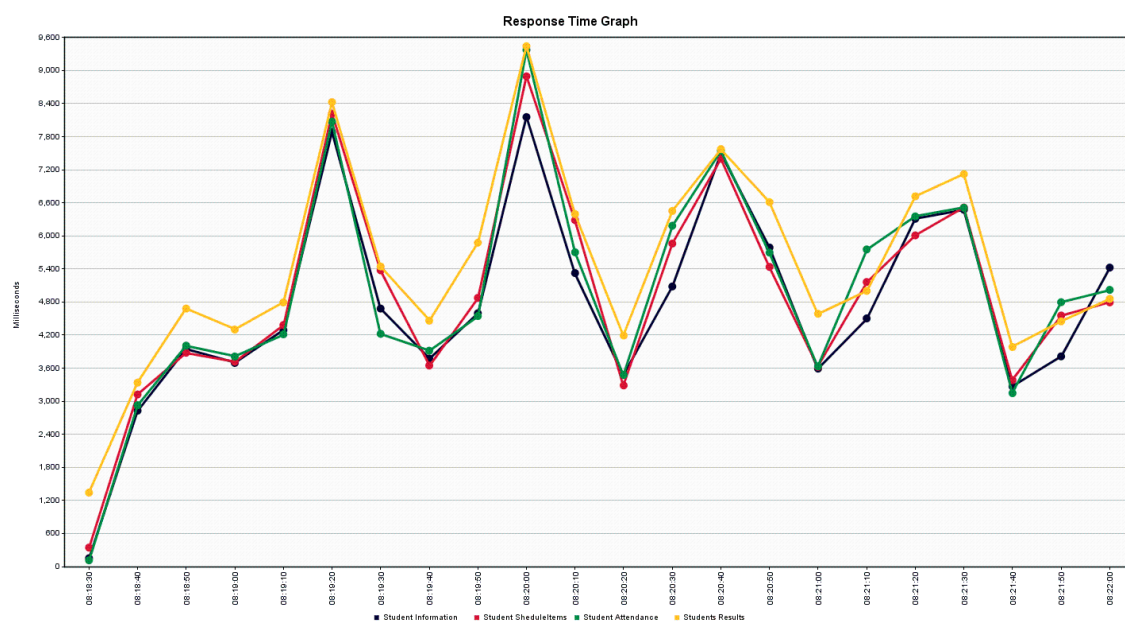
Obrázek 26: Agregáčnı graf učitelé - RaspberryPI, synchronizace 5 učitelů a 50 žáků součastně



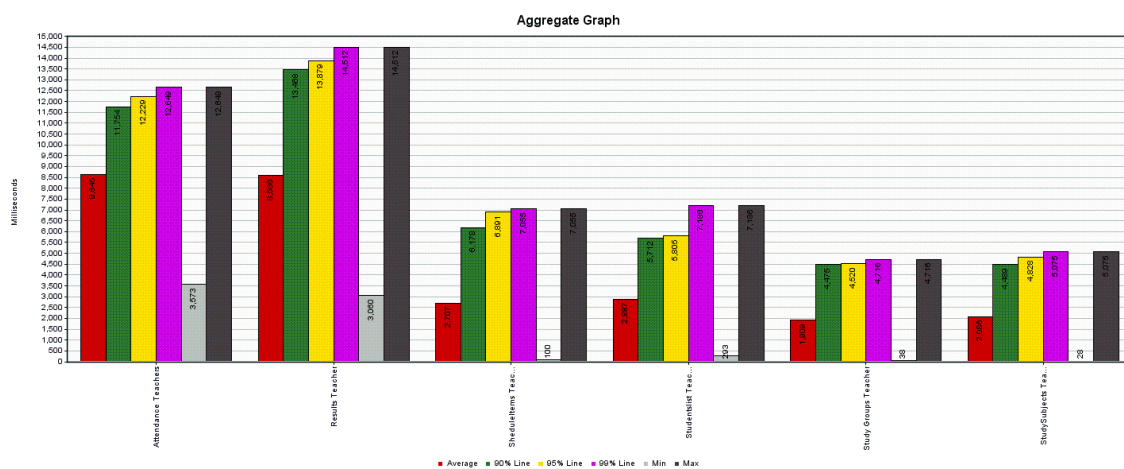
Obrázek 27: Graf průměrná odezva učitelé - RaspberryPI, synchronizace 5 učitelů a 50 žáků součastně



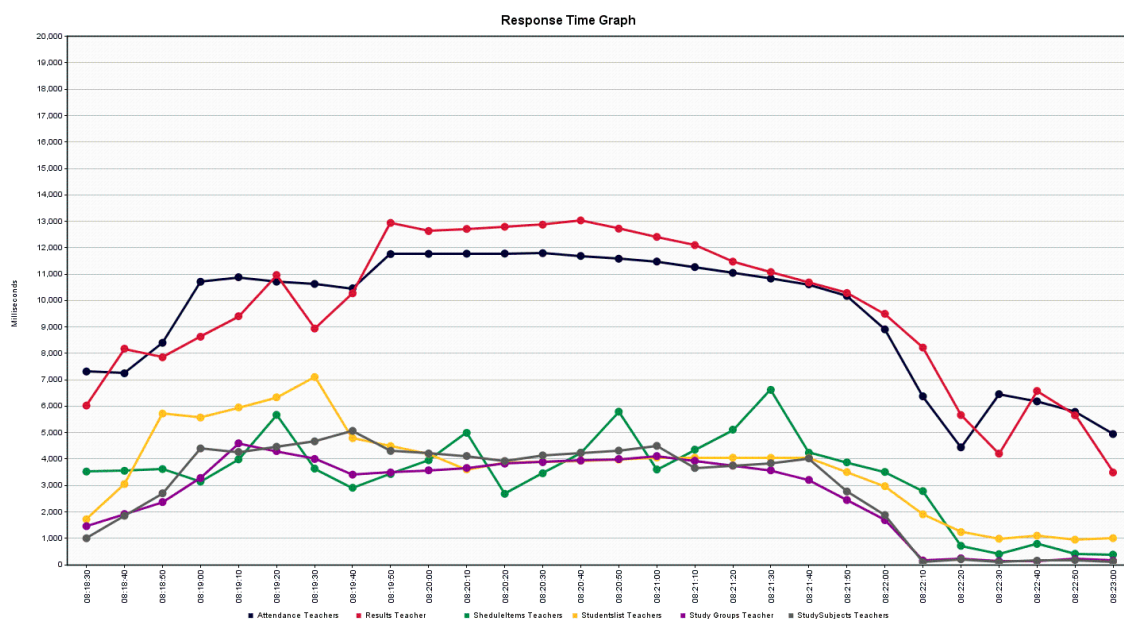
Obrázek 28: Agregáčnı graf řáci - Aspire One NetBook, synchronizace 5 učitelů a 50 řáků současně



Obrázek 29: Graf průměrná odezva řáci - Aspire One NetBook, synchronizace 5 učitelů a 50 řáků současně



Obrázek 30: Agregací graf učitelé - Aspire One NetBook, synchronizace 5 učitelů a 50 žáků současně



Obrázek 31: Graf průměrná odezva učitelé - Aspire One NetBook, synchronizace 5 učitelů a 50 žáků současně